AD-A047 930 MITRE CORP MCLEAN VA METREK DIV F/G 9/2
DISTRIBUTED PROCESSING APPLIED TO THE FLIGHT SERVICE STATION MO--ETC(U)
AUG 77 T B FOWLER DOT-FA69NS-162
UNCLASSIFIED MTR-7576 FAA/RD-77-161 NL

1 OF 3
AD
A047930

ADA047930

9

(18) (19)

Report No. FAA/RD-77-161

(6)

Distributed Processing Applied to the Flight Service Station Modernization Program

(10) T.B. Fowler

The MITRE Corporation

METREK Division

McLean, Virginia

(14) MTR-7576

(11) August 1977

(12) 203 p.

(15) DOT-FA69NS-162

Document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161.

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION

FEDERAL AVIATION ADMINISTRATION

Systems Research & Development Service

Washington, D.C. 20590

409 890

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FAA-RD-77-161		
4. Title and Subtitle: Distributed Processing Applied to the Flight Service Station Modernization Program		5. Report Date: August 1977
		6. Performing Organization Code
7. Author(s): T.B. Fowler		8. Performing Organization Report No.: MITRE Tech.Rpt. 7576
9. Performing Organization Name and Address: The MITRE Corporation, Metrek Division, McLean, Virginia		10. Work Unit No. (TRAIS)
		11. Contract or Grant No.: DOT FA 69 NS 162
12. Sponsoring Agency Name and Address: Federal Aviation Administration, Systems Research and Development Service, 2100 2nd Street S.W., Washington D.C. 20591		13. Type of Report and Period Covered: Feasibility Study
		14. Sponsoring Agency Code: ARD-404

15. Supplementary Notes

16. Abstract

This report is an investigation of the feasibility of using minicomputers in a distributed processing configuration for the Flight Service Station modernization program. It is based on studies of existing distributed processing configurations, many of which were visited by members of the MITRE/FAA distributed processing study team; on information supplied by vendors; and on an analysis of two sample configurations taking into account available data concerning Flight Service Station (FSS) functions. Distributed Processing, as applied to the FSS modernization program, is analyzed with respect to cost, degree of risk, and expected level of performance. Recommendations are given based upon analysis, research, and visits to existing distributed processing installations.

17. Key Words	18. Distribution Statement
Distributed Processing Flight Service Station Minicomputers	Document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	203	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

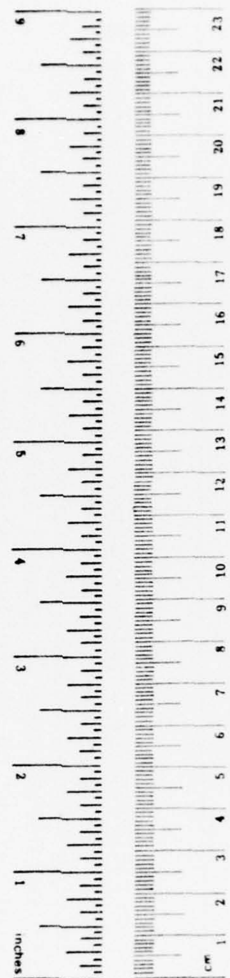
 409 890

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

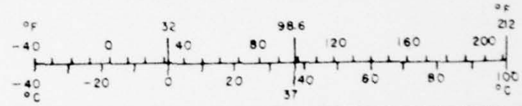
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in ≡ 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



SUMMARY

This report is an investigation of the feasibility of using minicomputers in a distributed processing configuration for the Flight Service Station modernization program. It is based on studies of existing distributed processing configurations, many of which were visited by members of the MITRE/FAA distributed processing study team; on information supplied by vendors; and on an analysis of two sample configurations taking into account available data concerning Flight Service Station (FSS) functions.

During the course of the investigation, it became apparent that while distributed processing is widely discussed and advocated, implemented systems are relatively few in number, and none has the load or complexity anticipated for an FSS Hub. Minicomputer software also emerged as lagging behind in development relative to hardware. However, at least one vendor does offer in packaged form the hardware and software necessary for distributed systems such as the FSS, though to date this vendor has not assembled a multiprocessor system as large as that needed for an FSS. The sample FSS designs are based on this hardware, and are configured as a load sharing system and as a function sharing configuration. Results of an analysis of these configurations indicate that the functions

of an FSS could be handled by them, with comfortable reserve capacity. A summary of the performance of these systems under peak loading conditions is shown in Table 1.

On the basis of information gathered in the report, the following conclusions may be drawn: (1) Distributed processing systems are not yet generally available as off-the-shelf items; (2) System software development of minicomputers lags behind their hardware development--of special concern is the apparent lack of availability of tested multiprocessor operating systems; (3) Considerable raw computing power is available from minicomputers, enough to handle the FSS functions; (4) Existing minicomputer software, though not as powerful as that available on large mainframes, is adequate for writing FSS application software; (5) Properly designed distributed mini-systems with good hardware have potential advantages with respect to cost, maintainability, expansibility and flexibility; (6) There is risk in using minicomputers for the FSS application, but there appears to be no technical reason which would preclude implementing such a system; (7) Procurement of premium grade hardware is essential to minimize risk.

Accordingly, MITRE recommends not barring distributed mini-systems from being bid for the FSS modernization program, provided that

TABLE 1

PERFORMANCE SUMMARY FOR RECOMMENDED FUNCTION SHARING AND LOAD SHARING CONFIGURATIONS, 1995 PEAK HOUR LOAD

(Response Time in Seconds)

CATEGORY	FUNCTION SHARING CONFIGURATION	LOAD SHARING CONFIGURATION	LARGE MAIN FRAME
1. Route Briefing, First Page	2.664*	2.683*	1.02
2. Local Briefing, First Page	.343	.255	.155
3. Flight Plan Processing	.800	.615	.330
4. Aircraft Contact Processing	.184	.198	.126
5. Miscellaneous Messages	.173	.183	.145
6. Remainder of Route Briefing	5.568	5.39	2.42
7. Remainder of Local Briefing	1.899	1.524	.785

* Mean drops by 1.2 - 1.5 seconds if route processing begins as soon as route is entered; see Section 4.3.1.

adequate safeguards are taken to ensure procurement of high quality hardware, that proper testing procedures are available to ensure reliable and adequate software performance, and that special attention is paid to securing a good multiprocessing operating system.

CONCLUSIONS

General

Distributed processing systems of the size and scope required by the FSS project are not yet available as off-the-shelf items, nor have any of the complexity envisioned for an FSS Hub been assembled. However, the hardware and software technology exist and have been assembled into a single package by at least one vendor. The largest system to date is about 50-60% of the estimated size of an FSS Hub, but does not handle the type of load anticipated for the FSS (dynamic data base, random demand, real-time operation, etc.)

The distributed processing systems assembled thus far have largely been custom affairs, optimized for the unique requirements of the particular application. All of those who have built such systems believe in the future of the concept, so long as too much is not demanded of the distributed system. There are jobs (as explained in Section 1) which are not suited for distributed processing, and there appears to be general agreement that large mainframes with high CPU speeds will never be totally replaced by distributed systems.

It is also important to bear in mind that the raw power of today's premium minicomputers (e.g., DEC 11/70, DG Eclipse) surpasses that of second generation mainframes (e.g., IBM 7094) and approaches that of

many third generation machines (e.g., IBM 360/65). Indeed, a benchmark run by General Dynamics suggest that in some respects they approach the power of some fourth generation machine.

Unfortunately however, software development for the minicomputers has not kept pace with hardware development, and while real time multitasking operating systems are available, they do not in general have the variety and scope of support packages or other system software, such as compilers, offered on the large mainframes. Typically, real time operating systems available off-the-shelf support a single processor and interprocessor communications only, and with the exception of TANDEM, not multiprocessing in the broad sense. Some operating system software development will probably be required no matter which system is chosen.

The jobs required for the FSS application do appear to be within the scope of present day minicomputers, and costwise the hardware needed for the job is significantly less than for a comparably powerful mainframe. There does not seem to be any reason to assume that minicomputers could not handle the Flight Service Station data processing load given the response time and other performance parameters as set forth in the specification.

Summary of Distributed Processing System Advantages

1. Lower cost for equivalent computing power.

2. Potentially higher reliability if correct system architecture employed, and premium quality hardware used throughout. This equipment is available but the specification must be very carefully written to insure its procurement. Reliability is something which must be designed into a system, and cannot be tested, patched or repaired into it.

3. Maintainability can be facilitated if hardware is correctly designed, i.e., is very modular. This, like reliability, is not an automatic feature of distributed systems.

4. Expansibility of mini-systems is a great advantage if the system software and overall architecture are such that interprocessor overhead does not reduce the power of the aggregate. A well-designed mini-based system should allow for considerable expansion capacity (on the order of 100% is not too much to expect).

5. Flexibility in configuration can enable the FSS Hub to cope more easily with unanticipated changes in service requirements and demands.

Summary of Distributed Processing Disadvantages

1. Distributed systems of the size and scope required are not available off-the-shelf.

2. Automatic reconfiguration hardware is available from only one vendor.

3. Benchmarking a multiprocessor system will be difficult if part or all of the system must be developed. Benchmarking efforts up to this point have been directed toward large mainframes.

4. Data recording for event reconstruction may pose some problems in a distributed system due to the fact that all the information which must be logged may not be available on one machine. Techniques such as combining of logging tapes off-line, etc., will probably have to be employed. These problems can most likely be overcome, but will require special software.

5. Lack of comprehensive, well-debugged software support packages, utility programs, and compilers. However, these features are available in varying degrees of refinement on most minicomputers.

6. Multiprocessor Operating Systems are not generally available off-the-shelf.

Estimate of Risk by Area

The MITRE/FAA Study group estimated the risk associated with distributed processing on an item-by-item basis, classifying each as low, medium or high. This Table is reproduced as Table 2.

Final Conclusions

The conclusions of this study are in basic agreement with those of the joint MITRE/FAA Study team.

1. Minicomputers have adequate processing power to handle the FSS application.

2. Distributed processing with solid hardware and good system architecture has advantages over mainframes for certain types of jobs.

3. Software requirements of the FSS system pose the greatest risk at this juncture. Some requirements, such as automatic reconfiguration, are a risk with mainframes as well. Though progress in the software area is difficult to predict, distributed processor manufacturers seem to be concentrating on developing software which is commensurate in power with their hardware, and many risk areas identified in this study may be eliminated within the time frame of the FSS procurement. That, however, cannot be guaranteed. Further, certain software (e.g., operating system for real-time multiprocessor applications and reconfiguration) is applicable to a relatively small segment of the near-term potential

TABLE 2

SUMMARY OF DISTRIBUTED PROCESSING RISKS BY AREA

	RISK		
	HIGH	MEDIUM	LOW
HARDWARE			
STORAGE (MEMORY AND DISC)			X
SPEED OF PROCESSOR			X
ERROR DETECTION			X
ERROR CORRECTION			X
SOFTWARE			
HIGHER ORDER LANGUAGES		X	
OPERATING SYSTEM SINGLE PROCESSOR			X
OPERATING SYSTEMS MULTIPLE PROCESSOR	X		
DEVELOPMENT TOOLS		X	
SYSTEM			
PERIPHERAL			X
THROUGHPUT			X
AUTOMATIC RECONFIGURATION	X		
COMPARABLE SYSTEM APPLICATION	X		

distributed processing marketplace, and as a result may receive little near-term industry attention.

4. The wide range of minicomputer equipment and performance dictates that a comprehensive system benchmark be performed at some stage in the procurement.

5. Premium grade hardware (CPU and peripherals) should be procured to minimize the risks associated with reliability, performance, maintainability and schedule problems. Attempted use of low quality hardware could result in catastrophic increases in cost and extremely poor performance.

6. The cost for a typical Hub system can be expected to range from about \$1.5M to \$1.7M for the type of configurations considered here, assuming high quality hardware used throughout.

RECOMMENDATIONS

On the basis of the analysis done in this report, and the information gathered for it, the following recommendations may be made:

1. Allow distributed minicomputer systems to be bid for the FSS modernization program, but make special provisions for the development of multiprocessing operating system.

2. Write the specification so as to ensure procurement of high-grade hardware.

3. Have comprehensive testing and benchmarking procedures available to permit proper evaluation of systems bid.

With respect to the analysis of sample configurations, further work should be done to refine those parameters which had to be estimated in order to carry out the analysis.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Contd)

TABLE OF CONTENTS
(Conc)

LIST OF TABLES

LIST OF TABLES
(Continued)

LIST OF TABLES
(Continued)

LIST OF TABLES
(Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Concluded)

1. INTRODUCTION

The term 'distributed processing' today has many meanings in the data processing field, ranging from geographically dispersed intelligent terminals to colocated tightly-coupled multiple mini- or micro-computers. In this report a distributed processing system is considered to be a loosely coupled, colocated set of minicomputers identical in type, with all peripherals necessary to perform the tasks of a Flight Service Station Hub. The purpose of the study is to analyze the feasibility of using such a system for the FSS application, with particular emphasis on the level of performance that can be expected from a properly designed system, the degree to which hardware and software essential to its operation are commercially available, and the overall system cost.

1.1 Types of Distributed Processing Systems

Distributed processing networks may be divided into four general categories:

1. Hierarchial, consisting of a powerful main frame, fed by successively less powerful smaller machines, down to the terminal level, with each job ascending in the hierarchy only as far as necessary to obtain the resources it needs for execution (See Figure 1-1).

2. Star, consisting of one central computer (mini or mainframe) linked in radial fashion to two or more other computers, which are controlled by it. The central machine performs all scheduling and task assignment functions, and may in addition handle a portion of the processing load. It usually has all or most of the data base. (See Figure 1-2.) There is no communication between satellite machines without intervention of the central computer.

3. Ring, consisting of several machines linked by a one-way circular path, along which all messages between machines must travel. Usually the individual processors retain a large degree of autonomy. (See Figure 1-3.)

4. Global Bus, consisting of two or more processors linked by a common bus line which, being a node, has no directional data flow; any machine can communicate directly with any other. (See Figure 1-4.) The individual machines are usually very autonomous.

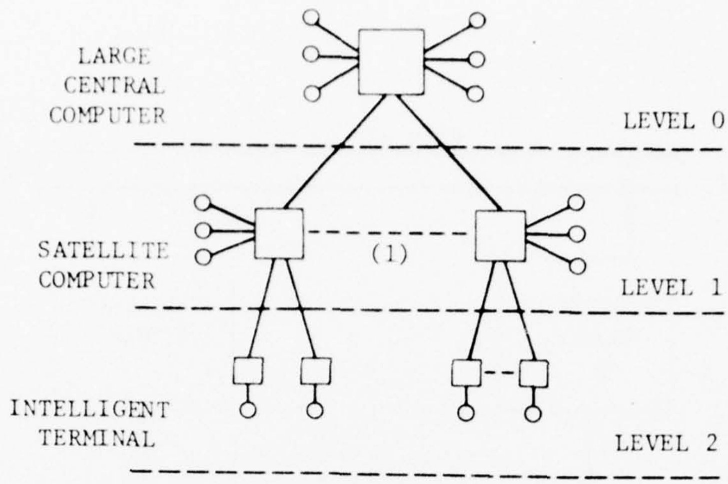


FIGURE 1-1
HIERARCHICAL NETWORK

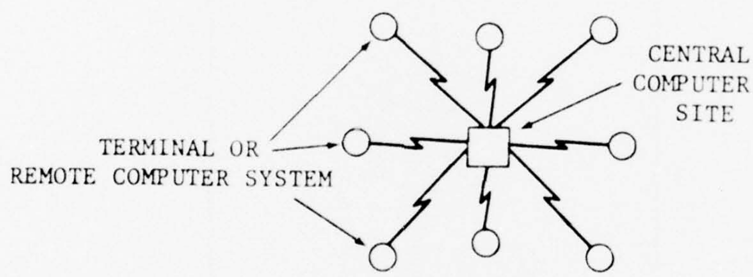


FIGURE 1-2
STAR NETWORK

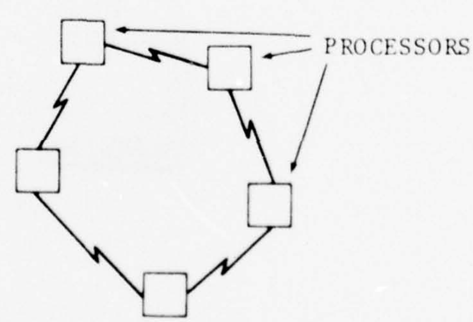


FIGURE 1-3
RING NETWORK

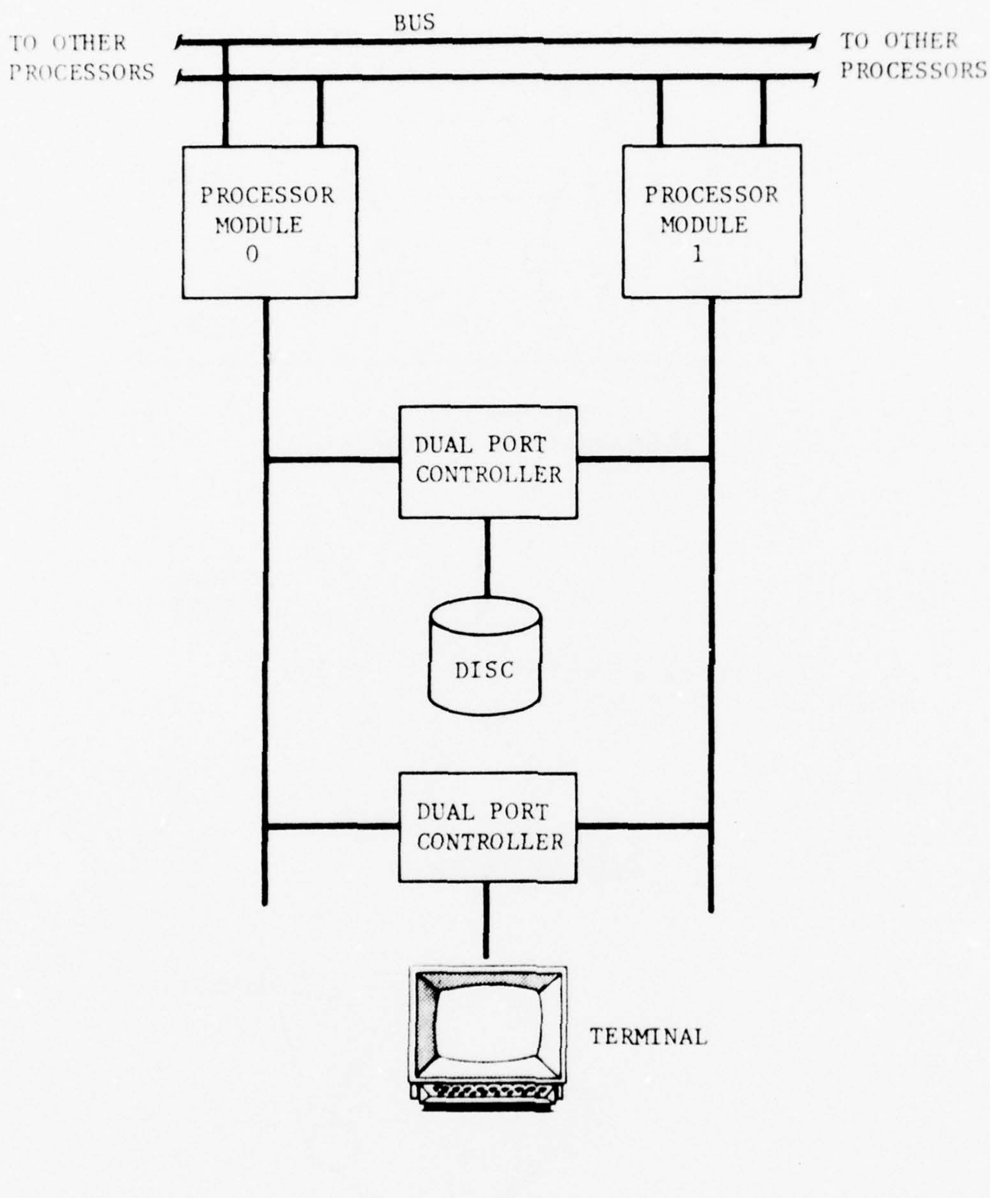


FIGURE 1-4
GLOBAL BUS NETWORK

Examples of the latter three types will be discussed below in Section 3.1. No hierarchial systems are considered here because large main frames are excluded from this study.* Hybrid types, of course, also exist.

1.2 Differences Between Distributed Mini Networks and Large Mainframes

The reasons for employing distributed processing configurations rather than large main frames must be considered. First, with regard to throughput, it is clear that m minicomputers each capable of n instructions per second will collectively be able to execute nm instructions per second. If the minicomputers cost less than one large main frame with the same (nm) raw computing power, cost will be a reason favoring the distributed configuration. But merely assembling a group of minicomputers and interconnecting them by means of an interprocessor link will not necessarily provide the same task throughput as the large main frame regardless of the raw minicomputer power available. The reason is quite simple: many tasks require serial execution of a large number of instructions, and cannot be broken down into tasks running in parallel. Clearly feeding

* For a description of a hierarchial network employing a 370/158, a PDP 11/45, and several smaller machines, see 'A Hierarchical Network', by R. L. Ashenhurst and R. H. Vonderohe, Datamation, Vol. 21, No. 2 (February, 1975), pp. 40ff.

such a job into a group of mini's will result in one being assigned the job, while the others do nothing. Obviously in this case the large main frame, with its higher execution rate, will complete the job much faster. However, in the case of short run-time, I/O bound jobs, which do lend themselves to processing in separate machines, the mini system could be as fast or possibly faster than the large main frame, which would have to handle the jobs using a time-slicing algorithm or pre-emptive-resume queuing scheme. The time required to complete a task, however, will depend on I/O bandwidth, operating system design, load, queuing, etc. But insofar as the task load consists of short run-time jobs, especially I/O bound, the distributed mini architecture is a candidate for replacement of a large main-frame if there is a significant cost difference, which there often is.

This situation has been recognized by designers and proponents of multi-mini systems. Dr. William Wulf, whose system is discussed in Section 3.1.3, says:

> By now the data processing community has become sophisticated enough to understand that there are no panaceas. Thus, although there are many advantages to multiple mini-computer systems, such systems are not ideal for all applications and they pose new problems which must be solved if these new architectures are to realize their full potential.

Perhaps the most obvious applications for which the multi-minis are ill-suited are the large "number crunching" ones, involving a great deal of floating-point arithmetic. Minis, multi or not, frequently do not have floating-point hardware, and even those that do are constrained by their small word size to implement such operations as slow, multiple memory reference instructions. They therefore fall short of large scientific processors both in basic cpu speed (by a factor of 2-5) and memory bandwidth (by a factor of 2-8), and these factors may go much higher for specific machines (like CDC STAR's pipelined floating-point processor or the IBM 360/85 with its cache memory). However, commercial data processing applications fall well within the mini's range of practical utility, since it is intended to perform byte manipulation and I/O with considerable efficiency...

Obviously, bit rates do not tell the whole story. In exchange for the speed inherent in parallel processing, the user receives the problem of task-decomposition--that is, the subdivision of an application program into pieces which can be executed in parallel and therby fully use the power available in the system. Fortunately for the user, he is not required to solve this problem at the outset; he can choose to build his program initially in the traditional, sequential way and rework it later. Obviously, his initial program will not run faster, but the system can still execute 16 such "temporarily sequential" programs in parallel, with no real loss in throughput over a multiprogrammed uniprocessor or batch system. Eventually, as the user designs an effective parallel decomposition, he can rebuild his program to reap the benefits of the architecture. *

* William Wulf and Roy Levin, 'A Local Network', Datamation, Vol. 21, No. 2 (February, 1975), pp. 49.

Second, a distributed architecture may lend itself to more modular implementation of functions, thereby effecting a potential increase in reliability if the operating system is capable of detecting failures and initiating automatic reconfiguration. The required redundancy is much less expensive than in the case of a large mainframe (which would necessitate duplication of a very costly CPU), and may in addition provide a degree of protection against multiple failures. It is important to note, however, that without proper hardware and proper system architecture, and in particular without some type of automatic reconfiguration, the distributed mini system could have a higher failure rate and lower availability due to a larger number of single points of failure. Distributed processing does not guarantee higher reliability.

Finally, Auerbach's general comment on the state-of-the-art in computer networking is interesting:

> An analysis of the literature of networking, such as that made by AUERBACH while developing its new multivolume looseleaf subscription service, "Distributed Systems," reveals that while
>
> - Most networks in use now are hierarchical,
> - The literature on networks is predominantly about distributed networks.

> What explains this apparent dichotomy? The answer is that hierarchical networks can be built today with off-the-shelf products from the major main frame manufacturers. Distributed networks cannot. They are only supported by minicomputer manufacturers like Hewlett-Packard and Digital (although Digital's DECNET cannot yet be fully implemented). From the large scale general-purpose computer point of view, distributed networks remain the stuff of advanced computer science courses and the province of DP buffs with large ideas and larger budgets. Hierarchical networks are highly utilitarian, concerned with fueling the engines of everyday commerce.*

Nevertheless, research conducted for this report indicates that the situation today is rapidly changing, and overall the picture is considerably brighter for certain applications of distributed processing. The Flight Service Station does appear to be one which can be adapted to a distributed architecture, and two sample configurations are analyzed in detail below, Section 4.

---

* Auerbach Computer Technology Reports, Sec. 050.0000.305, 'Network Design: Introduction to Computer Networks', pp. 5.

2. DESIRABLE FEATURES IN A DISTRIBUTED PROCESSING SYSTEM FOR THE FSS APPLICATION

The maximum load for an FSS Hub is summarized in Appendix A; the functions, number of machine instructions, and disk accesses for each are summarized in Appendix B.

The FSS baseline system function specification also requires fail-safe operation, which is defined as follows:

1. Error detection by the computer program.
2. Error isolation by the computer program.
3. Switching of the failed unit out of the system by the computer program to effect isolation of the failed unit.
4. Switching of a redundant unit into the system by the computer program to replace the failed unit.
5. Automatic resumption of operation.

There are various other general requirements concerning maintainability and reliability which should also be met by any distributed processing system.

Based on the above and other considerations, the desirable features for a distributed system architecture are listed and discussed below.

2.1 Hardware

2.1.1 Memory

Both memory size and type will be critical in determining the performance of any distributed computing system. Important features are as follows.

2.1.1.1 Capacity

The amount of memory required per machine will vary with system architecture, but based on previous estimates and on the experimental system at MITRE, at least 256K bytes per machine will be required, and for certain configurations up to 1M may be needed when allowance is made for reserve and expansion.

2.1.1.2 Type

Although either major type of memory (core or semiconductor) will function, semiconductor has some potential advantages in lower dissipation, less space required, and in future years, lower cost.

2.1.1.3 Error Detection

Clearly a desirable feature for fail-safe and/or fail-soft

operation is error detection in the memory. This is, in fact, almost indispensible for reliable operation.

2.1.1.4 Error Correction

A step beyond error detection, this feature permits the memory to correct single bit errors and continue operating with no system degradation. By reducing failures that require immediate maintenance action it improves system availability.

2.1.1.5 Type of Addressing

Memory may be addressed either directly using a fixed number of bits containing the binary address, or using two groups of bits, one being the direct address within a small area of memory, usually a page, and the other group a number reference to an entry in a table (called the 'map') containing the physical starting address of the page. There are potential advantages to a mapped system in that mapping permits dynamic reallocation around failed memory space.

2.1.2 Multiprocessing

Since any distributed system must be capable of interprocessor communication, there is a requirement for hardware to provide for high speed communications between processors. This hardware should permit DMA transfers of at least 100K bytes/second.

Since more than two processors will likely be involved, the hardware should not be of such type that if one processor fails, communications between other processors are impaired.

2.1.3 Reconfiguration

Since reconfiguration is a system requirement, clearly any computers selected must either offer it off-the-shelf, or it will have to be developed. Since development of such a capability, both hardware- and software-wise, is likely to be difficult and costly, this is a particularly desirable feature to have available as a standard item.

2.1.4 Machine Speed

This is not strictly speaking a feature as much as characteristic of all machines, and clearly for the FSS application, the higher the speed the better the machine, other things being equal. However, machines which look the same on paper, i.e., whose average instructions execution rates based on some standard mix are equal, may due to operating system or compiler inefficiencies turn out to be radically different in actual performance. As an example, a benchmark run by General Dynamics on a group of machines, including several under consideration for this project, showed that two of them, nearly equal on paper, differed by a factor of 30 in time to execute the test program. Hence, this characteristic must be regarded with caution.

2.2 Software

2.2.1 Multiprocessing

Since any distributed processing system by definition involves some degree of multiprocessing, it is desirable to have system software available from the vendor which supports multiprocessing to the maximum degree possible, for example some dynamic allocation of system resources.

2.2.2 Languages

Given that the FSS programs are to be written in a higher level language, it is most desirable that one or more standard languages be available on the machine selected, powerful enough to handle both numeric calculations and alphanumeric (string) manipulations. Development and debugging of compilers is expensive and time consuming, and should be avoided if at all possible.

2.2.3 Automatic Reconfiguration

This feature is taken in conjunction with hardware required to implement it, and the same remarks apply.

2.2.4 Communications Software

This package would be useful for communications with the outside world, but given the special nature of front end (terminals) and back end (NADIN) communication of the FSS, special application programs may have to be written anyway.

2.2.5 Data Base Management System

Most DBMS's are intended for applications such as on-line inventory control, and their usefulness for the FSS application may depend on the DBMS's flexibility and overhead cost. This study assumes that a special type of data base management for weather data will be employed. DBMS's are only now becoming available on minicomputers and their reliability may not be adequate, or at least demonstrable. If sufficiently flexible and reliable, however, and without a high penalty in overhead, a DBMS could be most useful in handling part of the FSS data base.

2.3 Survey of Minicomputers Deemed Suitable for the FSS Application and Their Features

Table 2-1 is intended to be a representative though not exhaustive survey of minicomputers available for possible use in a distributed processing configuration for the FSS program. Included in the table are the features discussed above which are deemed important to the FSS application. Information was taken

TABLE 2-1

COMPARISON OF OFF-THE-SHELF HARDWARE AND SOFTWARE FEATURES OF MINICOMPUTERS

	MEMORY					MULTI-PROCESSING			
MACHINE MANUFACTURER AND MODEL	CAPACITY (BYTES)	TYPE	ERROR DETECTION	ERROR CORRECTION	MAPPED OR DIRECT ADDRESSING	HARDWARE CAPABILITY	TECHNIQUE	SOFTWARE SUPPORTED	MAX NO OF PROCESSORS
TANDEM	512K 2M(1/78)	SC	YES	YES	MAPPED	YES	DUAL BUS	YES.	16
DATA GENERAL ECLIPSE C/330	512K	SC	YES	YES	MAPPED	YES	BUS	LIMITED	15
DEC PDP 11/70	2M	CORE	YES	NO	MAPPED	YES	COMPUTER TO COMPUTER INTERFACE	YES	NA
INTERDATA 8/32	1M	CORE	YES	NO	DIRECT	YES	SHARED MEMORY	NO	14
HP 21 MX	2M	SC	YES	NO	MAPPED	YES	GEN. PURPOSE INTERFACE	NO	NA
MICRODATA 3200	256K	SC	YES	NO	MAPPED	YES	GEN. PURPOSE INTERFACE	NO	NA
VARIAN V-76	2M	SC	YES	NO	MAPPED	YES	GEN. PURPOSE INTERFACE	NO	NA
TI 990/10	2M	SC	YES	YES	MAPPED	N/A	N/A	N/A	N/A
GENERAL AUTOMATION 16/80	256K	SC	NO	NO	DIRECT	YES	COMPUTER TO COMPUTER INTERFACE	NO	4
HARRIS SLASH/7	512K	CORE	YES	NO	MAPPED	YES	COMPUTER TO COMPUTER INTERFACE	NO	NA
MODULAR COMPUTER SYSTEMS MODCOMP IV	512K	CORE	YES	NO	MAPPED	YES	SHARED MEMORY OR COMPUTER TO COMPUTER INTERFACE	LIMITED	12 (SHARED MEMORY)
SYSTEMS ENGINEERING LABS SEL-32	1M	CORE	YES	NO	MAPPED	YES	INTERBUS LINK OR SHARED MEMORY	NO	NA

ABBREVIATIONS: SC = SEMICONDUCTOR
NA = INFORMATION NOT AVAILABLE

TABLE 2-1 (CONT'D)

COMPARISON OF OFF-THE-SHELF HARDWARE AND SOFTWARE FEATURES OF MINICOMPUTERS

		RECONFIGURATION				
MACHINE MANUFACTURER AND MODEL	HIGHER ORDER LANGUAGES	MANUAL	AUTOMATIC	APPROXIMATE MACHINE SPEED	COMMUNICATIONS SOFTWARE	DATA BASE MANAGEMENT SYSTEMS
TANDEM	COBOL (9/77)	YES	YES	420K IPS	YES	YES
DATA GENERAL ECLIPSE C/330	ALGOL, BASIC FORTRAN	NA	LIMITED	700K IPS	YES	NO
DEC PDP 11/70	BASIC, COBOL, FORTRAN	NO	NO	400K IPS	YES	YES
INTERDATA 8/32	BASIC, COBOL (6/77) FORTRAN	NO	NO	650K IPS	YES	NO
HP 21 MX	ALGOL, BASIC FORTRAN	NO	NO	92K IPS	YES	YES
MICRODATA 3200	-	NO	NO	260K IPS	NO	NO
VARIAN V-76	BASIC, COBOL, FORTRAN	NO	NO	430K	YES	YES
TI 990/10	BASIC, COBOL, FORTRAN	N/A	N/A	N/A	N/A	N/A
GENERAL AUTOMATION 16/80	BASIC, FORTRAN	NO	NO	350K IPS	YES	NO
HARRIS SLASH/7	BASIC, COBOL, FORTRAN, SNOBOL	NO	NO	550K IPS	NO	YES
MODULAR COMPUTER SYSTEMS MODCOMP IV	BASIC, FORTRAN	NO	LIMITED	430K IPS	NO	NO
SYSTEMS ENGINEERING LABS SEL-32	FORTRAN	NO	NO	530K IPS	YES	NO

ABBREVIATIONS: SC = SEMICONDUCTOR
NA = INFORMATION NOT AVAILABLE

from Auerbach Reports, Datapro Reports, and vendor documentation when available. It is believed to be accurate but may not in all cases be completely up-to-date. There is no implication that MITRE judges these machines to be the only ones suitable for the FSS application.

There are a few points to be made with respect to the features listed. First, no machines having less that 256K bytes of memory capacity were considered, since this, in MITRE's opinion, is the absolute minimum for any likely FSS distributed configuration. Most vendors offer optional battery supplies to save semiconductor memory in the event of a power failure.

Multi processing was considered to be software supported only if the operating system viewed the entire network as an available resource, and had a comprehensive scheduling algorithm for allocating system resources as needed. Simple recognition by the operating system of other computers was not considered to be software support, though the higher lever of multiprocessing may not be required for the FSS.

The higher order languages listed are those claimed by the vendor. Only 'universal' languages were considered; vendor designed languages are not included.

Reconfiguration was considered to be available if what the vendor claimed was approximately what has been defined as automatic reconfiguration for the FSS. If the vendor claimed some reasonable subset of the features 'limited' was placed in the column entry.

The machine speed given is approximate only, based on the following instruction mix (when information available).

Load/store	15%
Fixed add	25%
Logical AND	25%
Shift	15%
Floating Pt. Add	10%
" " Multiply	5%
" " Divide	5%

This information is intended only as a rough guide, and should not be used to judge the various machines. An accurate measure of machine speed will require a benchmark.

The final two columns, again, were completed based on vendor information, no independent verification of these software packages is available.

3. EXISTING DISTRIBUTED PROCESSING SYSTEMS

As part of the effort to determine the feasibility of distributed processing as applied to the FSS system, visits were made to several sites which have a reputation for pioneering this field. Reports on several others were obtained, all with a view to learning their problems and obtaining recommendations from them concerning distributed processing in general. Part of this effort was directed to learning whether any existing sites are of a complexity comparable to that anticipated for the FSS system.

The installations/sites visited or researched were as follows:

Non-FAA	FAA
Citibank, New York	DARC
Southern Bell, Atlanta	DABS
Carnegie-Mellon, Pittsburgh	ARTS III-A
Bell Labs 'Spider', Murray Hill, New Jersey	AWANS
JETS, Ottawa, Canada	
MITRE MTI System, Bedford	
Kansas State University	
DCS, University of California, Irvine	
Carter Associates, Cupertino, California	
MITRE TICCIT System	
ARPA Network	

3.1 Descriptions of Installations/Sites

3.1.1 Citibank, New York

This is one of the more widely known distributed processing systems. Citibank's application is a stock transaction system and mini's were chosen specifically to replace IBM-360's. The following points relevant to the FSS application may be made with respect to this system.

1. It is a load sharing configuration.
2. Split is along organizational rather than functional lines.
3. Response time is 3-5 seconds, but not deemed critical.
4. There are several identical systems, and the configuration of each consists of two Interdata 8/32 minis and 2 to 5 300 megabyte drives at each location; one computer on line; other switched in manually if first fails.
5. No auto reconfiguration.
6. No interprocessor communication.
7. No remote terminals.
8. Any intermachine transfers, e.g., new programs, are handled by physically carrying a tape.
9. An important factor in the choice of Interdata was its similarity to IBM with respect to programming. There was also a minimal benchmark performed on several machines, which consisted of determining if and how fast each could

run some sample COBOL programs in use at the time on Citibank's 360's.

10. Only time requirement is 72 hour SEC regulation on turnaround; if one system fails completely, work can be done on another; all systems in same building.

There was general agreement among those present at the briefing that this application is too dissimilar to the FSS system to permit any inference about the feasibility of using minicomputers for the FSS job.

3.1.2 Southern Bell, Atlanta

The minicomputer system installed by Southern Bell in Atlanta, however, is more closely related to the FSS application. Given in Figures 3-1 and 3-2 is a block diagram of the configuration, and in Tables 3-1 and 3-2 a summary of system downtime and availability.

The purpose of this system is to permit requests for service, repairs, installation, etc., to be entered from remote terminals spread over a wide geographical area. The requests are then processed by the system, which makes up work orders to be released at the appropriate time, and handles all accounting functions and record keeping associated with the job.

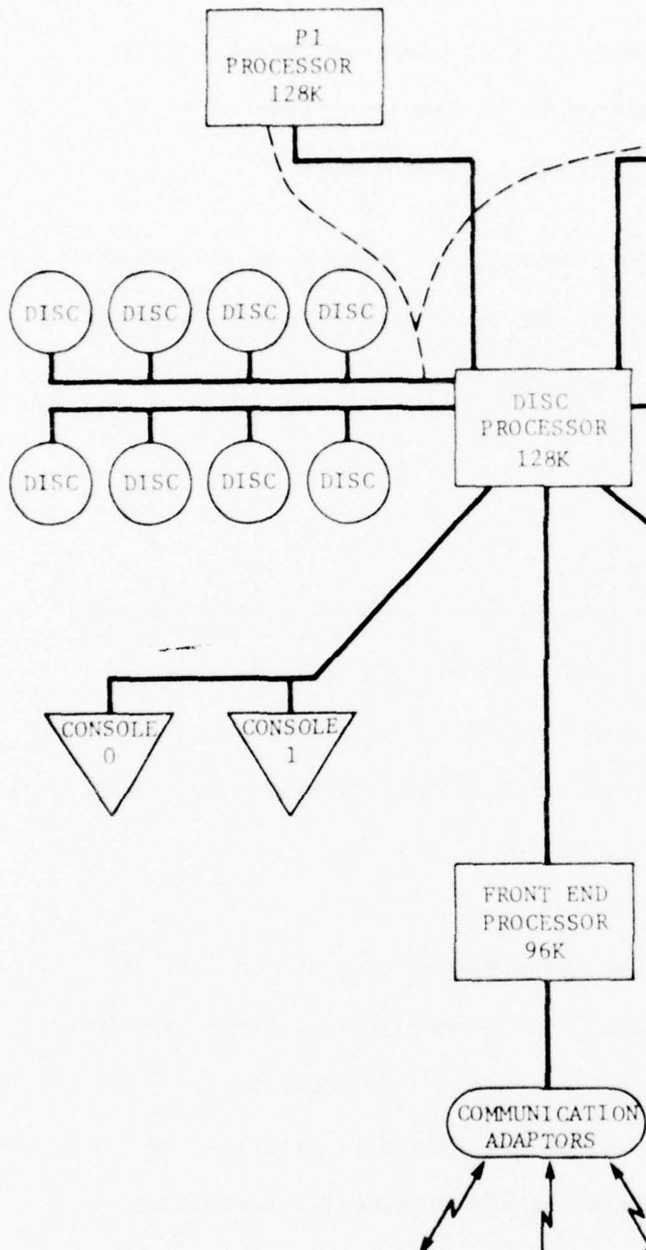


FIGURE 3-1
SOUTHERN BELL MULTIPROCESSOR SYSTEM

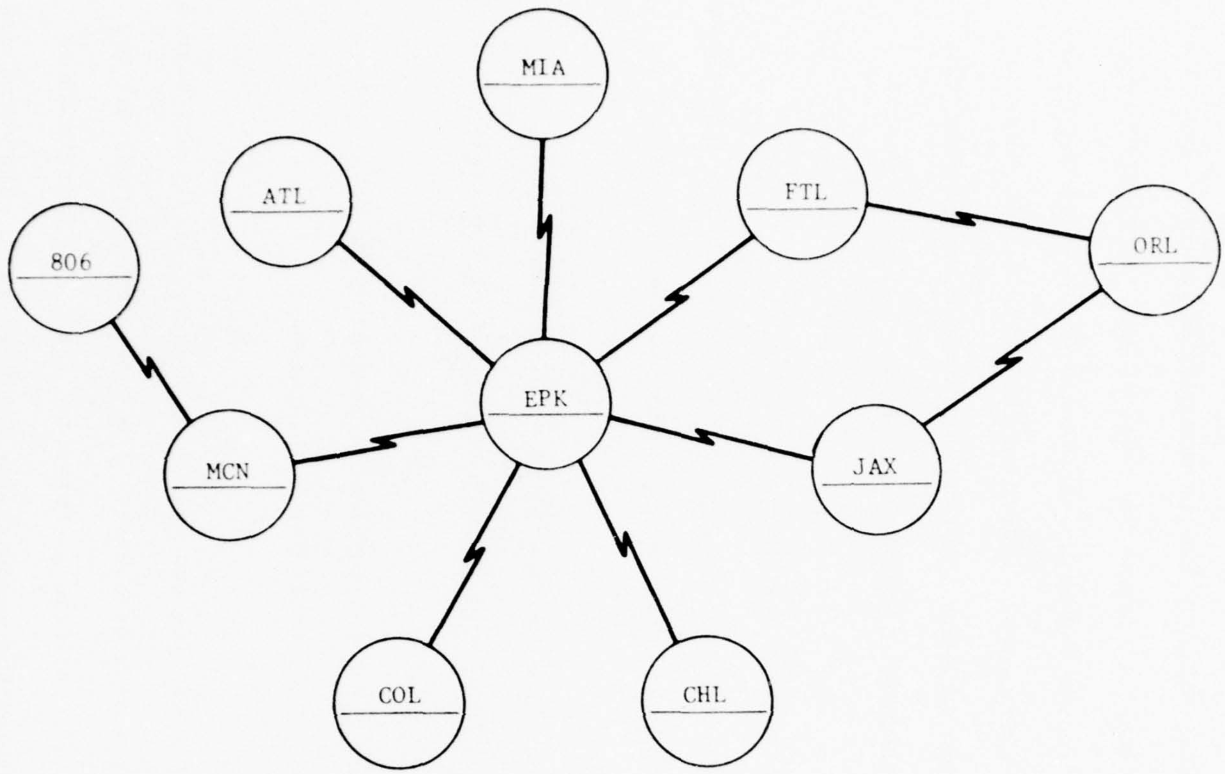


FIGURE 3-2
SOUTHERN BELL CORPORATE DATA NETWORK

The system does not have automatic reconfiguration, but in the event of a failure in either the disk processor or the front end processor, P1 or P2 can be manually switched in (by means of a bus switch) to assume the load of the failed processor. System up time is summarized in Tables 3-1 and 3-2 and as can be seen is quite good. According to Southern Bell, most of the down time is due to maintenance difficulties rather than inherent hardware or software problems.

The system functions as follows:

1. Service order typed in, entries checked for correctness.
2. Completed entry sent to disk processor, where it is logged.
3. Order then relayed to P1 or P2 processor, sent to distribution, held for acknowledgment of completion (up to one week).
4. When job done, completion message typed in, information relayed to all other parts of system.

Each computer has its own operating system, but all are basically slaved to the disk processor. The interprocessor channel was DEC equipment, modified by Formation Inc., the systems house which assembled the hardware for Southern Bell. The front end can handle up to 240 terminals, though presently the maximum

TABLE 3-1

SOUTHERN BELL DATA SYSTEM
DOWNTIME SUMMARY FOR 1976

08:00 - 1800

	% UPTIME	
08:00-18:00	8 SITES	10 SITES
JANUARY	97.6%	98.0%
FEBRUARY	97.1%	97.6%
MARCH	95.8%	96.3%
APRIL	98.6%	98.9%
MAY	98.0%	98.4%
JUNE	98.2%	98.6%
JULY	97.8%	98.3%
AUGUST	97.8%	98.2%
SEPTEMBER	98.5%	98.8%
OCTOBER	98.5%	98.8%
NOVEMBER	97.5%	98.0%
DECEMBER	98.4%	98.7%
1976 TOTAL	97.8%	98.2%
OBJECTIVE	99.4%	99.6%

TABLE 3-2

SOUTHERN BELL DATA SYSTEM
DOWNTIME SUMMARY FOR 1976
00:00 - 24:00

	% UPTIME	
00:00-24:00	8 SITES	10 SITES
JANUARY	98.4%	98.7%
FEBRUARY	98.4%	98.7%
MARCH	97.4%	97.7%
APRIL	98.7%	98.8%
MAY	98.9%	99.2%
JUNE	98.8%	99.0%
JULY	98.2%	98.6%
AUGUST	98.5%	98.8%
SEPTEMBER	99.0%	99.2%
OCTOBER	98.8%	98.9%
NOVEMBER	98.7%	98.9%
DECEMBER	98.2%	98.6%
1976 TOTAL	98.5%	98.7%
OBJECTIVE	99.8%	99.8%

connected at any installation is 115. Any response times over five seconds are considered poor; 95% of the responses are less than 10 seconds but some take as long as 90 seconds.

The operating system (exclusive of applications programs) was written in an assembly language developed by Southern Bell, and was operational in 18 months; but total development effort involved seven programmers for three to four years. The fact that the operating system was written in assembly language is apparently causing some problems; at present more core would help throughput and response time but would require major operating systems changes for support.

The system is similar to that envisioned for the FSS in that it is real-time, multiprocessor, reconfigurable, checks input, holds data for completion messages to be input later, and supports a large number of interactive terminals. It differs in that large volumes of data are not output to the entering position, response time is not regarded as quite so critical, and the data base is not being continuously updated.

3.1.3 Carnegie Mellon University - C.mmp, Pittsburgh

This installation was visited by a team from MITRE and the FAA, and described in an article in Datamation, February 1975,

pg. 47-50. C.mmp is an abbreviation for Carnegie-Mellon Multi-Mini Processor.

The system consists of 16 processors and 16 memory boxes, interconnected by a switch which is the central component of the system. According to Professor William Wulf;

> This switch allows any of the processors to access any of the memory boxes on each memory reference. The processors are not permanently attached to a memory box, rather, each time a processor wishes to access a particular memory, a temporary connection is established through the switch for that access. Sixteen separate processor-memory connections are possible simultaneously. Thus, unless two processors are attempting to access the same memory, 16 separate and simultaneous communication paths exist between processors and memory.*

A diagram of the system is shown in Figure 3-3.

The system can be configured into several smaller, independent systems; and this partitioning enables a type of fail-soft software architecture to be implemented, so that the system degrades gracefully in the event of a CPU or memory module failure. An offending unit is 'partitioned out' for maintenance, and the rest of the system continues to run; but there is apparently no way of assuring that a malfunctioning processor cannot contaminate all of memory; nor is there automatic detection of errors. Also, because the system has several potential

* Op. Cit p. 47

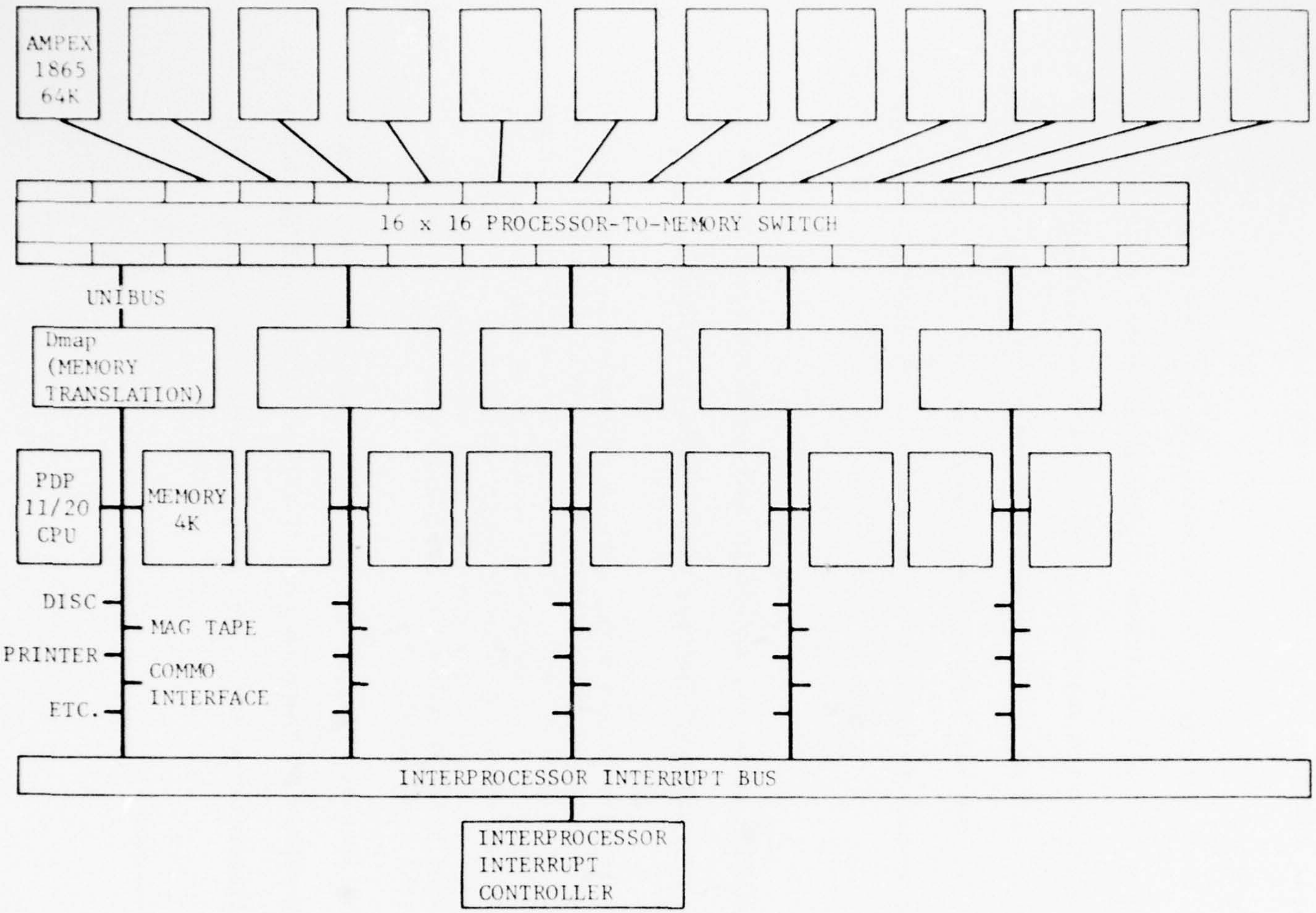


FIGURE 3-3
CARNEGIE-MELLON MULTI-MINI PROCESSOR

single points of failure (e.g., the switch and the interrupt bus), it would be very difficult to make fail-safe.

The C.mmp naturally has a custom-written operating-type system called a 'kernal', designed to provide maximum flexibility in the use of system resources. For the FSS application it is probably much too close to the hardware to be practical. There is no job control language, and the overall system is not transparent enough to be safe for or useful to the average programmer.

The authors, moreover, appear to have overlooked an important fact in queueing theory. They say with regard to response time:

> A multiprocessor with, say, 16 processors, can provide better real-time responses than a single processor which is 16 times as fast. Even though it clearly takes longer to service each individual real-time event, service for an event can usually begin immediately in a multiprocessor system, since more than one processor is available to respond to the (high priority) request for real-time service.*

What is involved here is the so-called 'scaling effect', which is the effect on response time of replacing one machine operating at m instructions per second with n machines operating at m/n

---

* p. 49

instructions per second. As it turns out, not only do response time and queueing time not go down, or even remain the same, they in fact go up by a factor of n.*

So the merits of this or any distributed processing system with respect to response time must be evaluated by a careful application of queueing theory to the particular design.

Overall, the C.mmp System is interesting, but there are major differences between this application and the FSS program as currently envisioned.

With respect to distributed mini-systems in general, Dr. Wulf discussed his experience with the MITRE/FAA study team which visited his site, and following is a quotation from their final report:

> Dr. Wulf indicated the following risks concerned with minicomputer distributed networks: no off-the-shelf operating system exists today, but the expertise is available; the need for a high-order language...; proficient data base management programs are not available; a floating point estimate of 20% may form the need for a 32-bit word machine; using mapping techniques in 16-bit machines may reduce efficiency by 30%; late recording of transient errors upon task completion was the biggest problem encountered;

* A. O. Allen, 'Elements of Queueing Theory for System Design,' IBM Systems Journal, No. 2, (1975), p. 167.

and functional program assignment to processors does not provide the best efficiency in a reduced mode of operation.

3.1.4 Bell Labs 'Spider', Murray Hill, New Jersey

'Spider' is a small packet-switched data communications system, designed to link a number (currently 11) of computers. The link is accomplished through a Terminal Interface Unit (TIU) at each computer, all of which in turn are connected by a high speed line to a computer functioning as a switch (see Figure 3-4). The purpose of this system is to interconnect the different types of computers being used at the Bell Labs facility, so as to provide better service to the users of the individual machines by allowing them to take advantage of the computing power available on other machines. The unique feature of this system is its 'Virtual Channel', whereby computers appear to be directly linked to each other while conversing, even though no assignment of physical equipment is made, and the machines may be operating at quite different rates. The heart of the virtual channel is the microprocessor based TIU, which replaces modems and data sets and permits handling of 64 full duplex data transmissions at one time. As Dr. Fraser, the system designer describes it;

> This means that the PDP-11/45 shared file system, for example, can carry on simultaneous conversations with several other computers even though it has only one TIU. In addition, the TIU provides buffering for one packet of data on each of its input and output paths. This allows terminal I/O

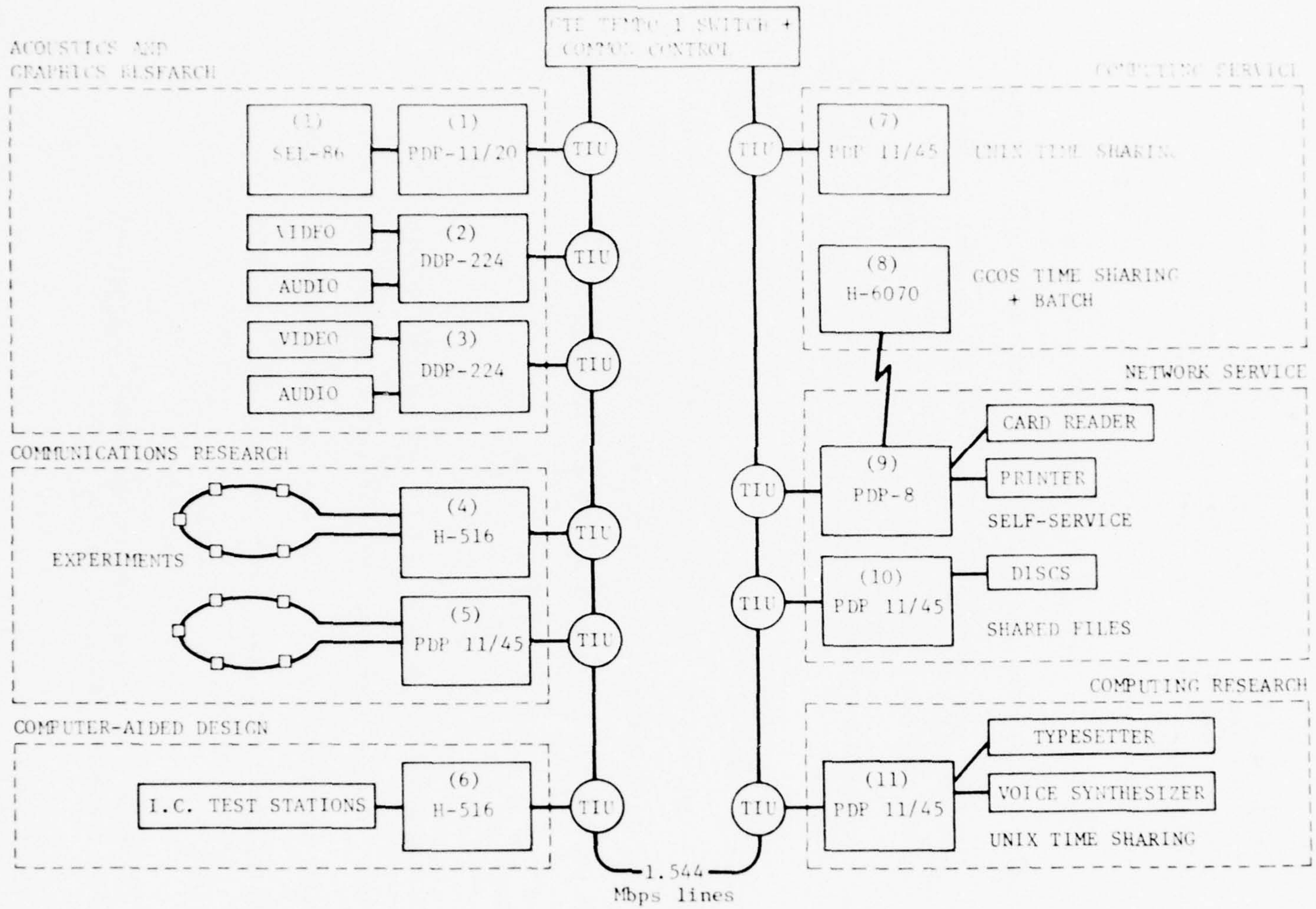


FIGURE 3-4
BELL LABORATORIES' 'SPIDER' NETWORK

> operations to proceed asynchronously at any speed up to the maximum the network can handle, and the TIU can retransmit the packet automatically when transmission error makes that necessary.*

Since the Spider network is not designed primarily as a distributed processing system, but as a computer-to-computer communications system, its primary relevance to the FSS program is in the area of intercomputer communication, in that it demonstrates one flexible, low overhead technique for linking a large number of machines.

3.1.5 JETS, Ottawa, Canada

'JETS' is an acronym for Joint Enroute/Terminal System, and is intended to gather and process radar data on air traffic, for presentation to controllers on display equipment.

A JETS installation consists of two central processors (one on-line, one hot standby), one tracking processor, one system control processor, and up to 32 display processors, each controlling one display (see Figure 3-5).

* Dr. A. G. Fraser, 'A Virtual Channel Network', Datamation, Vol. 21, No. 2, (February 1975), p. 52.

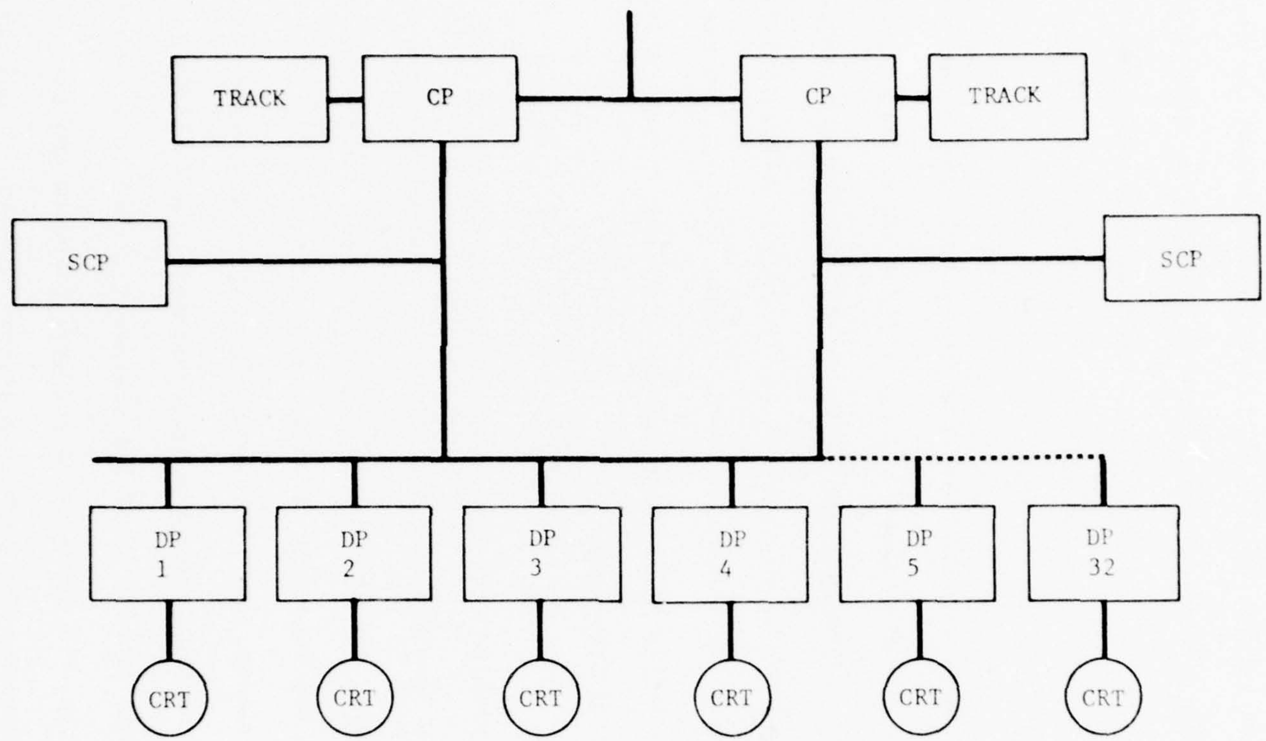


FIGURE 3-5
CANADIAN JETS INSTALLATION

The machines in the system are Interdata model 70. They were not chosen *per se*, but the Contractor who received the award chose them. There was some question at the time about the advisability of this selection, which appeared to be heavily influenced by price; and indeed it led to problems later.

At the time the project was undertaken, there was a desire to stimulate domestic industry and for this reason, as well as for reasons of reliability and cost, the procurement was restricted to minicomputers. Discussions on the project began in 1971, the contract was awarded in 1973, and the first system due in 1975. Presently delivery of the first system is expected in September of this year.

The main problem with the system up to now has, of course, been the delays involved with it. There are several reasons for these delays, the Canadian staff feels. First, there are problems with the Interdata 70 processor. It is restricted to 64K bytes of memory, had a poorly designed operating system, and (contrary to the Canadian's expectations) Interdata did not upgrade this machine later but instead switched to a new design. Consequently, the project has obsolete hardware and unsupported system software. The operating system currently in use represents a 70% modification of Interdata's RTOS.

Our understanding of the general feeling among the Canadian staff is that while minicomputers are quite suitable for large distributed processing systems, the following four constraints should be an integral part of system design:

1. Software must be upward compatible within the family of machines to facilitate using new hardware as it becomes available.

2. Off-the-shelf operating systems should be used.

3. Application programming should be done in a higher order language both for reasons of portability and maintainability.

4. Allowance should be made for memory expansion.

The Canadians have, in fact, taken their own advice in implementing a new system, called the Oceanic system, which uses two PDP-11/40's to support 24 terminals. Here, 80-90% of the code is in FORTRAN with the remainder in assembly language, and the operating system is DEC's RSX-11M.

A third (communications control) system uses a PDP-11/35, with programming again in FORTRAN.

Since the JETS system is not yet operational, no conclusions can be drawn as to its performance or practicality. The main benefit to be gleaned from a study of this system is that of learning some major pitfalls of multi-minicomputer system design.

The attitude of the Canadian Air Transportation Ministry toward distributed processing was summarized in the joint MITRE/FAA report:

> Although problems were encountered in the development of the JETS system, a commitment to minicomputers with more stringent software standards continues in future development systems. The low cost with higher reliability forms the basis for this approach.

3.1.6 MITRE MTI Multicomputer Network, Bedford

The purpose of this network is to do the data processing for GEODSS, which is itself a network of telescopes to search the night sky for distant man-made earth satellites. Detection is based on the relative movement of satellites compared to the fixed star background. A sensor is used to scan the star field, in raster form, and then after 10 seconds, the field is scanned again and the images compared for any telltale

indications of movement. The operation is described in detail in MITRE Report M76-201.

The amount of raw computing power required to implement the searching algorithm is enormous: approximately 9 million operations per second. Implementation by a single machine would have required either a CDC Cyber 176 (rated at 15 MIPS) or a UNIVAC 1100/40 (rated at 11 MIPS). An alternative strategy was to subdivide the task, and implement it on a group of minicomputers. The nature of the task lends itself to parallel processing, and this is how the final system was configured, using 10 minicomputers. A block diagram of the network is shown in Figure 3-6, and the data flow in Figure 3-7. The minicomputers used are Data General NOVAs, having an instruction rate (fixed add) of 1.2MHZ, with a bus transfer rate of 1.1M bits/second. Dr. Hiram Connell, designer of the system, summarizes the advantage of this hardware configuration:

> The net results of the division of tasks and data are that the storage associated with each computer is reduced to minicomputer dimensions, and processing speed requirements are reduced to manageable rates. Eight of the computers comprising this MTI system contain 16K words of 16-bit storage, and two have 24K words of storage. The computer instruction rates are a nominal 1.2MHz. This reduction is achieved by not requiring any one computer to contain an entire data set and by not requiring any computer to perform a complex processing task. Furthermore, each of the parallel processing branches contains an identical computer with identical software. Peripheral equipment for one master computer is all that is necessary

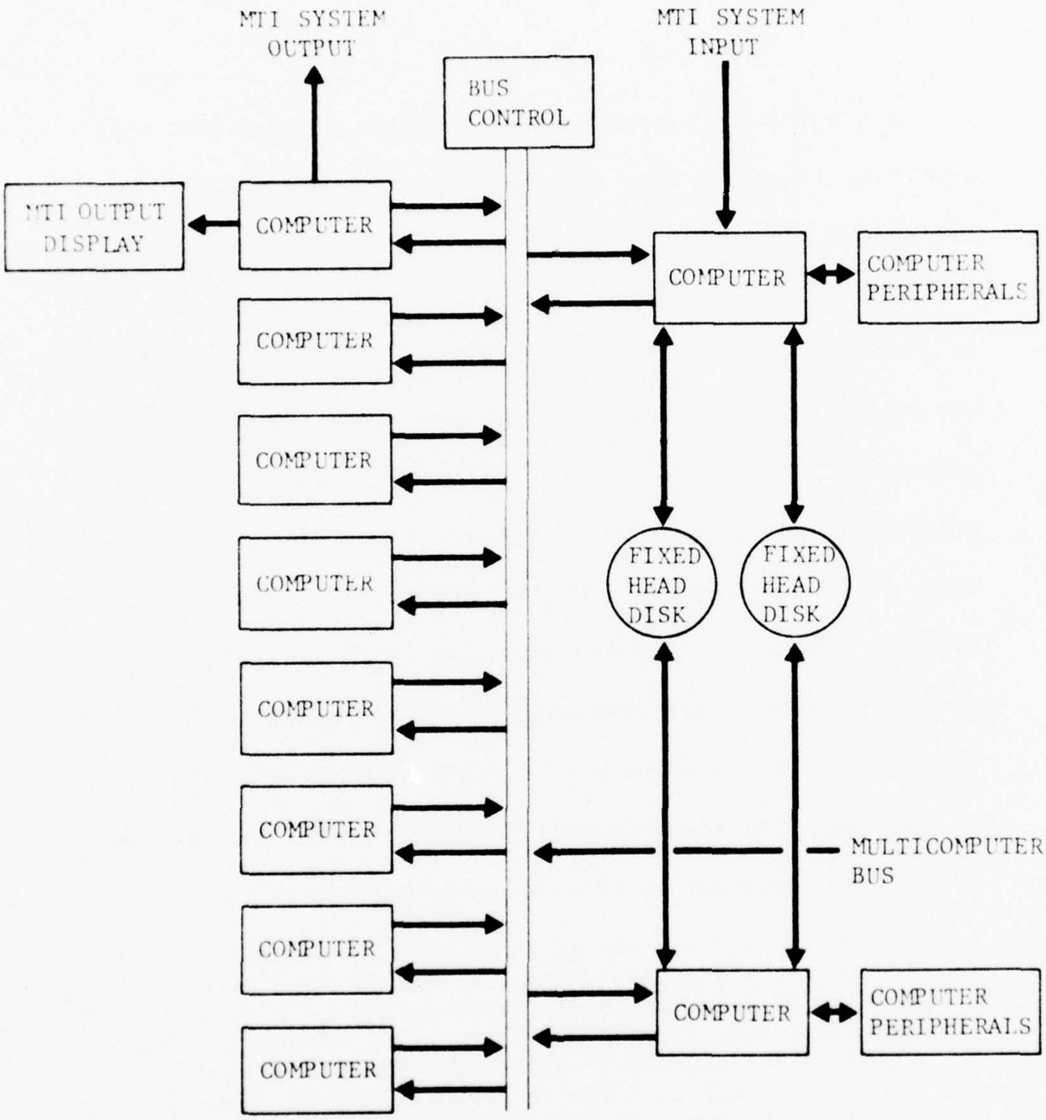


FIGURE 3-6
MITRE MTI MULTICOMPUTER NETWORK

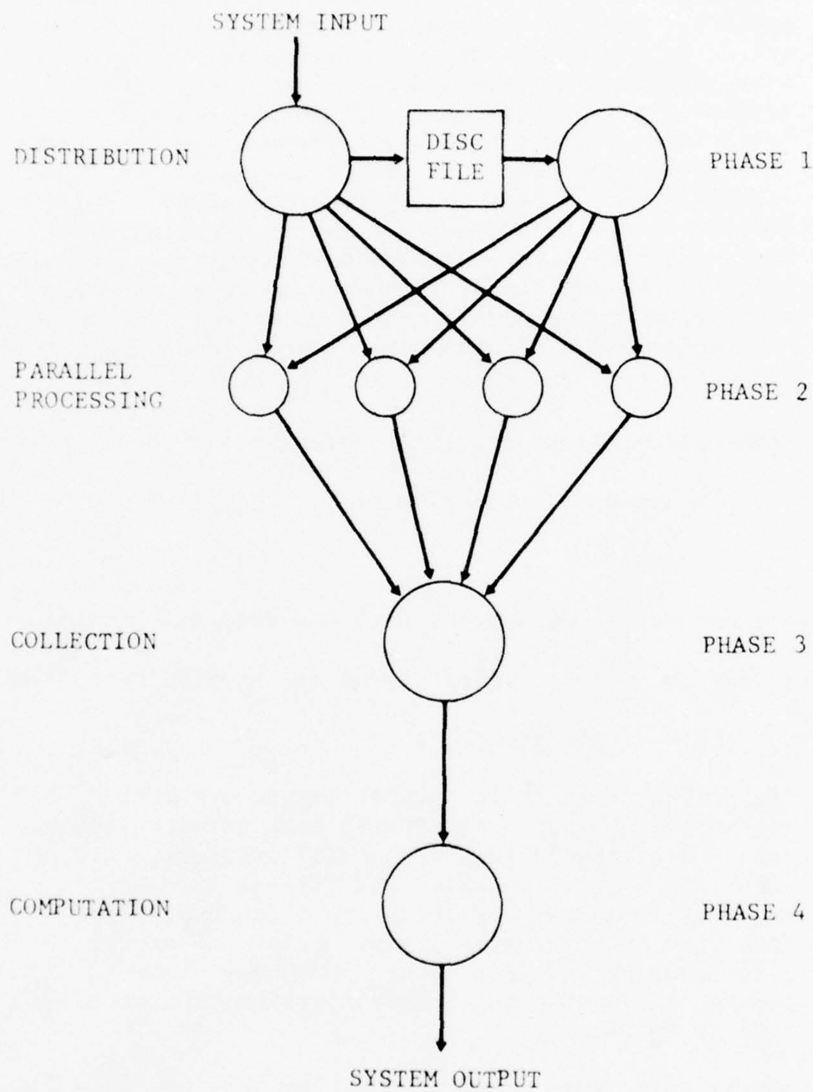


FIGURE 3-7
MTI DATA FLOW

> to develop software and operate the network. The entire software package for the specific MTI application, described here, is somewhat larger than would be required to implement the same process in one large computer; however, the modularity of the programs result in a lower cost and more error-free software development. Since each small program has well defined interfaces, modularity is nearly complete. The end result is that a form of enforced structured programming evolves in the development of the applications software. Each computer contains isolated program segments. Owing to the functional independence of each program segment, software development may proceed independently. For the snapshot MTI process, program development was rapid. . . .

The total hardware cost for this system is $200,000, or about 10% of the equivalent large mainframe.

Software and operating system, problems were also minimized by use of what Dr. Connell refers to as an 'implied executive'. As Dr. Connell explains it:

> If the requirements for proper inputs are strict within each computer program in each network element, then a distributed form of control exists; namely, each program will operate properly--as soon as it receives its data. By preparing a grand plan for data flow and approximately balancing the amount of processing required of each computer element before configuring the computer system, the executive is implied.

But clearly this type of architecture is only possible for jobs where the input and data processing loads are not only known, but are constant; it could not be used if the input or processing were governed by any type of stochastic process.

So despite its elegance and efficiency, this architecture is not suitable for the FSS application, which does deal with widely varying job loads, and inputs governed by stochastic processes.

3.1.7 Kansas State University

Dr. Paul Fisher, Chairman of the Kansas State University Computer Sciences Department, is in charge of a distributed processing developmental effort for the Department of the Army.

The Army laid down general guidelines for the project, among them two of great importance: (1) Network must be transparent to the type of processor used, and (2) operating systems of the various machines used could not be modified.

Since the system is not complete, only general information about it is presently available. Following is an exerpt from the MITRE/FAA distributed processing report dealing with this installation.

> The limitations placed on the network led to the selection of a microprocessor interface to a high speed, byte parallel, 10 megabite transfer rate bus. The network supervisor will reside in the microprocessors and will treat the minicomputers as resources. The network will be complete about midsummer and will consist of three nodes connected

through a communication interface--KSU 370/158, DEC PDP-11/70 and a cluster of minis at the KSU Computer Science Department. The cluster will be two Interdata 8/32's, one Interdata 7/16 and a Data General NOVA. The microprocessors will be Interdata 5/16s.

The OS overhead in the cluster was approximately 40%. The system programs were written in assembly language although the application programs were in standard COBOL.

Based on the definition of [FSS] system requirements, the following recommendations were made:

[Since] Most minicomputer problems are disk oriented, [the following should be used]:

· Core Memory
Parity Checking
32 Bit Word Size

... program [should] be task oriented and hardware assignment... based on system priorities and hardware availability. [The] big risk is not buying enough hardware to perform function. (Recommend 100% plus.) This is especially true if [it is] difficult to estimate ... future requirements.

The 32 bit word size recommendation comes about because of the floating point arithmetic required for route processing; however, this forms a small enough portion of the overall job that it should not be the driving force; and in fact the existing MITRE experimental PSBT System does not have 32 bit words and still adequately handles all arithmetic functions. The other recommendations will be discussed below.

3.1.8 DCS, University of California, Irvine

The DCS (Distributed Computing System) at the University of California, Irvine is under the aegis of Prof. David Farber, and is one of the better known distributed processing systems in operation today, and probably the best example of ring architecture.

The DCS installation was documented in Datamation, February 1975, and consists of three Lockheed SUE machines and two Varian 620i machines, none of which, according to Professor Farber, is exceptionally reliable, though all were up at the time of the MITRE/FAA visit. The machines are interconnected via a ring, the hardware for which was designed and built using TTL/SSI chips. There is no buffering of data by the individual ring interface; only a one bit delay used for synchronization. Traffic moves in only one direction around the ring, and there is a round-robin polling of the machines on the ring so that no one machine can saturate the ring with continuous transmission. Messages contain a source and destination code, a length indicator, and several bits which are set or not depending on whether the message was received and accepted. Obviously each ring interface represents a potential single point of failure, though to date (after 2-1/2 years) none has failed.

The software for the system, written largely in a higher level language developed for the purpose, is designed to be fail-soft rather than fail-safe, though the system does have fairly sophisticated methods for detecting errors and dealing with them, including saving the process environment, starting a test process, initiating another copy of the processor which failed, or merely suspending all activity until some explicit instructions are received from an operator. There is no automatic restart of programs, and limited reconfiguration capability. The system is designed for experimentation, teaching, and research, so only supports five terminals, though more could be added if desired.

A block diagram of the system is shown in Figure 3-8. There is no master operating system _per se_; the system is based on _processes_, not processors, with the goal of distributing not just hardware, but control and function. According to Prof. Farber:

> Each processor on the ring has a resident software system called the nucleus. The nucleus provides facilities for the scheduling of processes and for transmitting and receiving messages. Other system functions, such as resource allocation, device I/O, and file system services are provided by processes executing in the DCS. Because the nucleus is the only software absolutely bound to a particular processor, all other system services may be executed by any machine in the ring and can be accessed from any user processes through the message system. A process

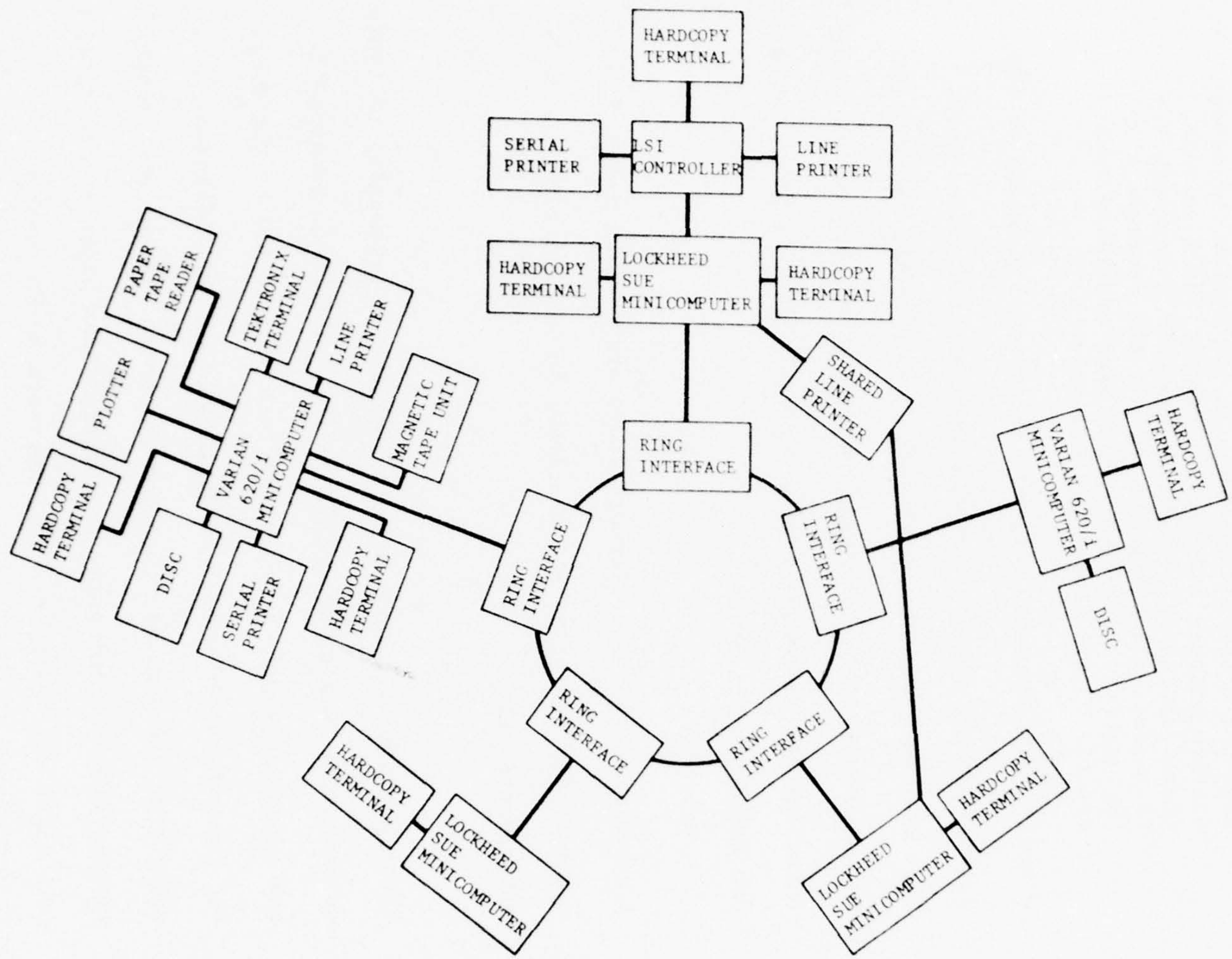


FIGURE 3-8
DCS RING AT UNIVERSITY OF CALIFORNIA, IRVINE

> requesting service addresses requests by name, it does not need to know where in the system the needed service process resides. A request for a service, issued at job initiation or whenever a new resource is needed, are recognized by all the processors in the system. Those processors with available resources in effect "bid" to supply the service.*

Hardware and software problems are detected by means of the interprocessor protocol (status bits set), and time-outs. Most overhead connected with this system is not attributable to the distributed processing configuration, but to the error detection and processor (not process) restart software required for fail-soft operation.

According to Professor Farber, there are systems being constructed for real-time applications, but apparently none is yet operational.

There would most likely be some problems in attempting to apply this architecture as it stands to the FSS system, because of the potential single point of failure represented by the ring. But the ring could easily be duplicated and if sufficient safeguards are designed into the system to prevent one runaway machine from contaminating the others' data bases or memories, this might well be a practical approach to FSS design.

* Prof. David Farber, 'A Ring Network', Datamation, op.cit., p. 46.

A detailed report on this system is available, including recommendations for future ring-based architectures.*

3.1.9 Carter Associates, Cupertino, California

The MITRE/FAA team visited one site having a TANDEM Computer System. Since the TANDEM system architecture and hardware appears to be suited to the FSS application and meets most of the stringent fail-safe requirements as well, the team wanted to learn about a customer's actual experience with the system.

Presently Carter Associates, which is a data processing house, uses three IBM 360's (2 65's and 1 40) in batch mode to provide services to their customers. These services include payroll, inventory control, order processing, etc. They decided that an interactive system might be more flexible and, if implemented with minicomputers, less expensive and potentially more reliable. They considered DEC and Data General machines, and were on the verge of buying an Eclipse for this application when TANDEM announced its system, which they decided to purchase instead. Their configuration consists of two processors, with about 30K of semiconductor memory per processor,

* The Distributed Computing Operating System, Lawrence A. Rowe, Technical Report #66, June 1975, published by the Department of Information and Computer Science, University of California, Irvine, California, 92664.

one disk, one tape drive, and several terminals. When their interactive data base development software is complete, they plan to expand their present configuration with additional processors and peripherals. Up to now they have written their software in TAL (TANDEM's own high level language, similar to ALGOL), but intend to convert to COBOL when released later this year.

Carter Associates is satisfied with their system, though its operation has not quite been non-stop. There have been approximately two hours of down time since the system was installed in August of 1976. This down time was attributed to software bugs (since fixed). Two hours of down time in 6 months corresponds to an availability of .9995.

3.1.10 MITRE TICCIT System, Washington

TICCIT (Time-shared Interactive Computer Controlled Information Television) is a minicomputer based distributed processing system dedicated to student instruction. It is a relatively small system computer-wise, (2 computers, 1 display generator), but is similar to the FSS application in that it handles loads whose distribution is Poisson, and the processing required for each transaction can vary over a wide range. In addition, the system logs all entries, does frequent checkpointing, and has

some automatic restart capability (though no automatic reconfiguration). It is capable of driving up to 128 terminals, each displaying up to 100K bits of information in 7 colors. Update time for completing changing a display is about 100 msec.

The original purpose of the TICCIT system was to demonstrate the feasibility of low-cost computer-aided instruction by employing minicomputer technology. The system is described in detail in two MITRE publications, M76-44, An Overview of the TICCIT Program, and M76-52, TICCIT System Specification. A generalized block diagram of it is shown in Figure 3-9, and a block diagram of the computer system as installed at two operational sites in Figure 3-10.

All TICCIT systems have four basic components: a main computer system referred to as the "main processor" and its associated data base; a terminal computer system called the "terminal processor" together with its data base; a display generator and refresh system; and student terminals composed of color TV receivers and keyboards.

The main processor, utilizing the TICCIT courseware data base, creates and assembles frames to be displayed on student

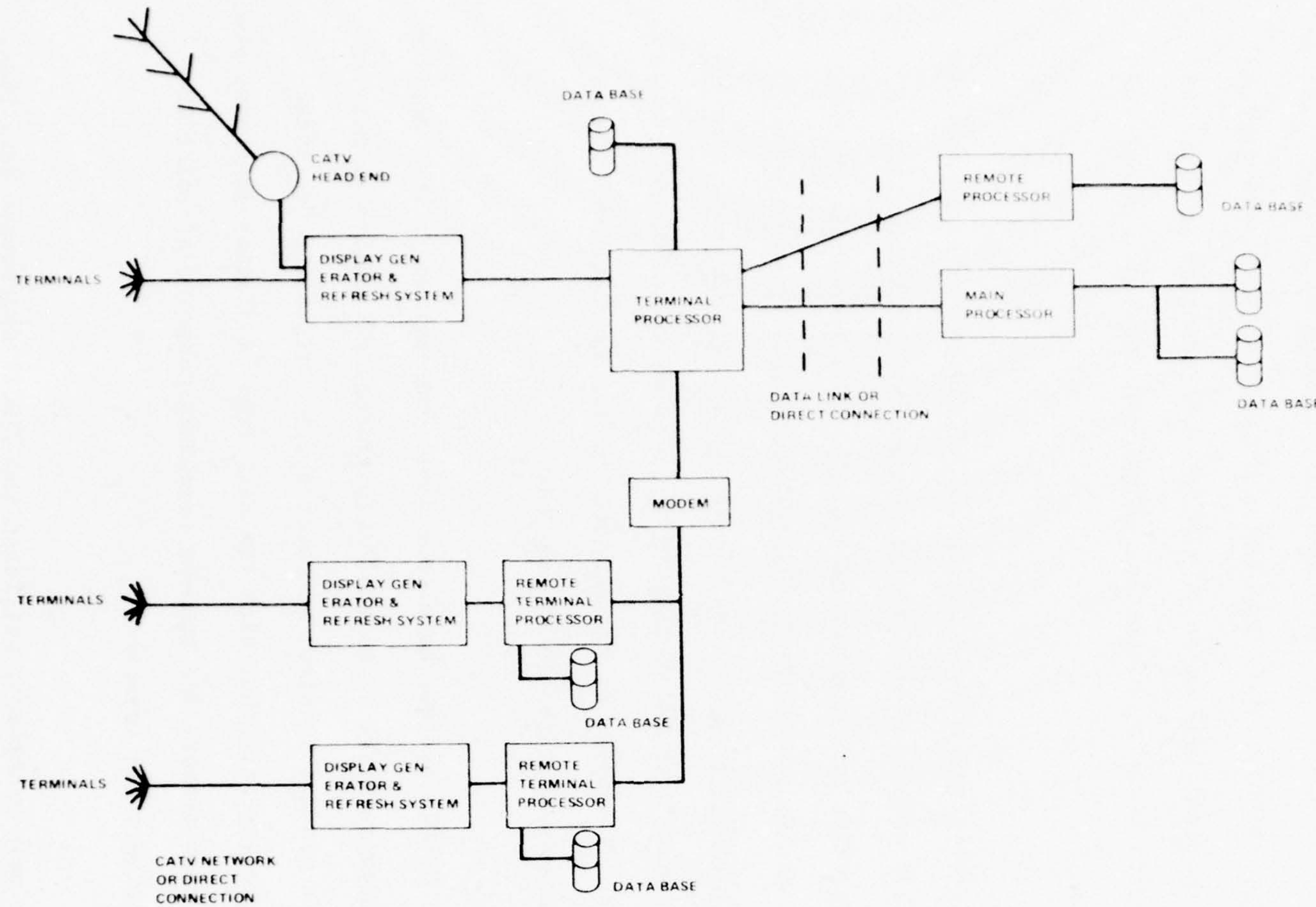
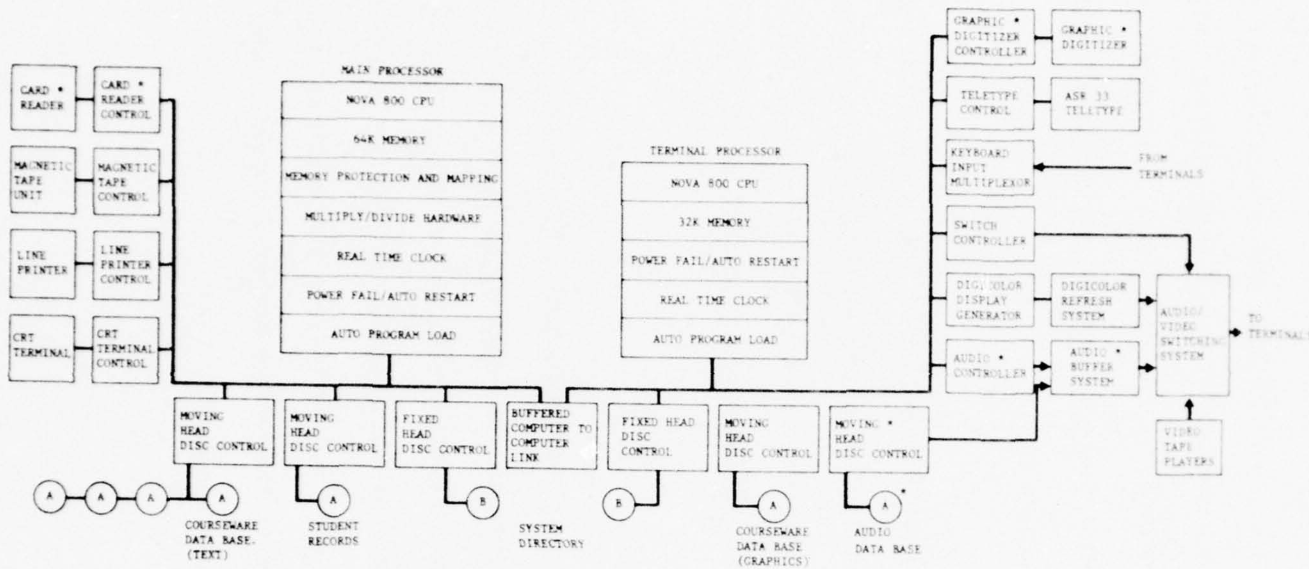


FIGURE 3-9
GENERALIZED TICCIT SYSTEM



A CENTURY 114 MOVING HEAD DISC MEMORY 12M WORD

B FIXED HEAD DISC MEMORY 128K WORD

* OPTIONAL

FIGURE 3-10
TICCIT COMPUTER SYSTEM

terminals as a function of course material and student response. Tasks of the main processor are diverse and relatively slow-paced, and include record keeping and student answer processing. The terminal processor performs all fast-reaction, highly-stereotyped functions; it interacts with the TICCIT student terminals, and does frame and audio outputting as well as keyboard input multiplexing.

One of the more unique features of the system is its display terminals. A terminal displays alphanumerics and graphics in seven colors, under computer control, as well as full-color videotapes. Up to 17 lines of 43 characters each may be displayed.

Graphic displays are constructed on a bit by bit basis on a grid of 204 elements in the vertical direction by 430 elements in the horizontal direction. The color of each half-character (i.e., 6 x 10 block of bits) may be individually specified. Full color videotapes add variety to the computer/student interaction and give the course author a powerful tool to dramatize difficult concepts.

To further augment the visual display, the terminal loudspeaker (or headphones) brings prerecorded messages under computer control.

Although the TICCIT system itself is not powerful enough to handle the FSS job, it does clearly demonstrate that a large number of interactive terminals can be driven by a distributed processing system which must process inputs from the terminals, access a large data base, and display responses to them quickly (average TICCIT response time, under full load, is better than 1 second). Furthermore, these terminals may be separated from the main processing unit by great distances.

3.1.11 ARPANET

The widely-known ARPANET is basically a packet switched communication system interconnecting a large number of existing machines. This type of 'distributed processing' is the geographically distributed kind, with which this report is not concerned. However, a multiminicomputer system is being developed to serve as a switching node for the ARPANET.*

---

* See 'A New Minicomputer/Multiprocessor for the ARPA Network', 1973 National Computer Conference, AFIPS Conference Proceedings, 1973, pp. 529-537.

3.1.12 DARC System

The DARC (Direct Access Radar Channel) System, designed by Raytheon as a partial backup for the 9020 Computers in the 20 ARTCC's, is a distributed processing system in the form of a star, but with a hybrid distribution of functions. It is implemented partially as a load sharing system (for display processing), and partially as a function sharing system (for central control functions). The purpose of the system is to permit display of radar data on PVD's in the event of a failure in the 9020's at any Center. A block diagram of the system is shown in Figure 3-11.

The system is capable of expansion to accommodate both the addition of radars and addition of display processors(DP's). Additional functions can be added to the existing Control Processor or by expanding the number of Control Processors. The system is not fail-safe, but it does have some redundant elements and is capable of manual reconfiguration.

The minicomputers used for processing are all Raytheon RDS-500 machines, which are somewhat smaller and less powerful than the FSS system will probably require (maximum memory capacity: 128K bytes). The operating system and the application software are written in assembly language, probably due in part to the

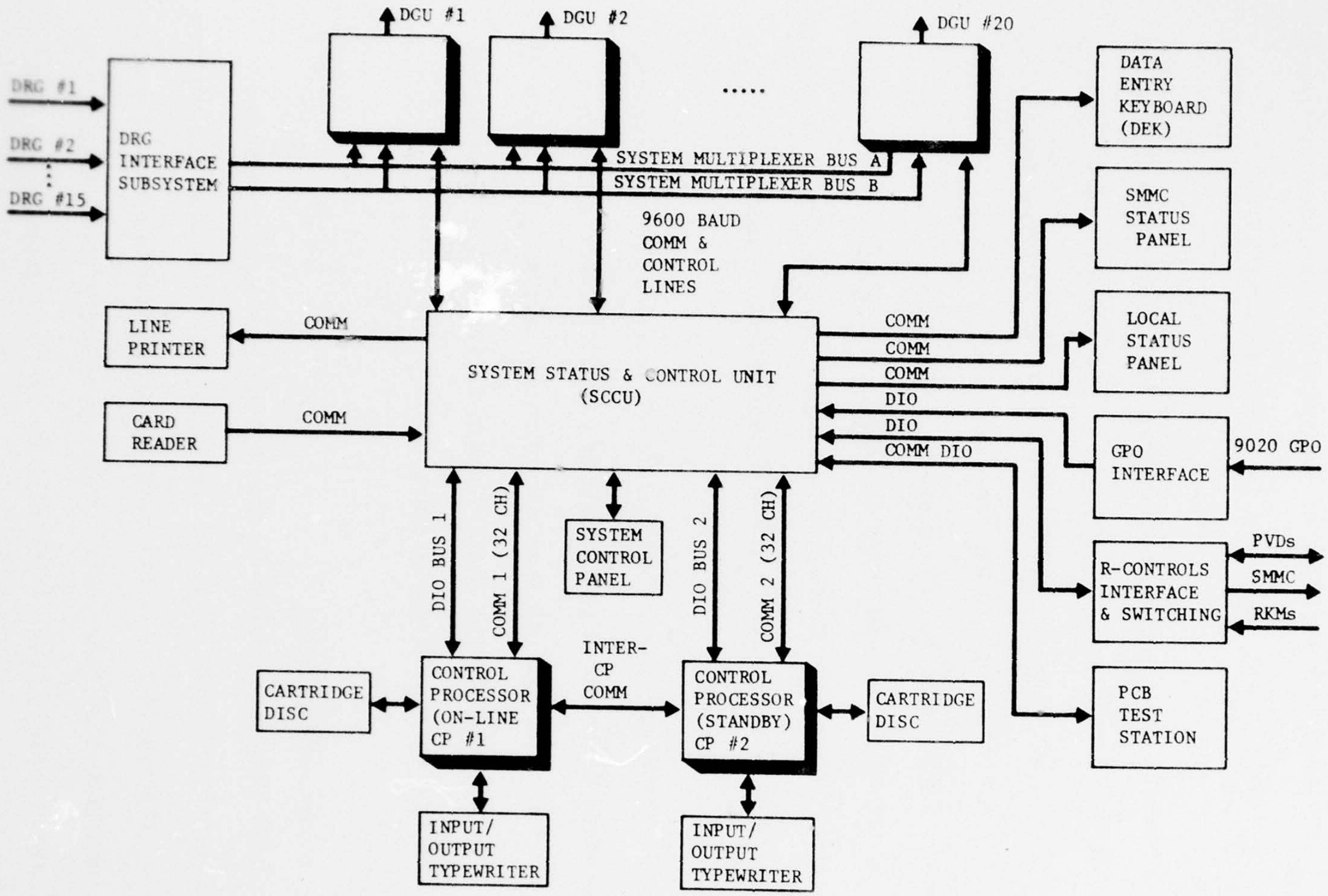


FIGURE 3-11
DARC SYSTEM ARCHITECTURE

limited memory capacity of the machine. The first system is scheduled to be delivered to NAFEC in March 1978.

Overall, this system demonstrates the feasibility of employing large numbers of minicomputers to perform a group of related tasks, but like the MITRE MTI system, is an ideal application for minis because of the inherent parallel nature of much of the processing, and the fact that the system load is not governed by the same type of stochastic process as expected in the FSS system. Hence, the DARC architecture cannot be considered directly applicable to the FSS system requirements.

With respect to the feasibility of large-scale minicomputer based distributed processing systems, Raytheon noted in a briefing to the FAA that such systems have been built for ATC use in Switzerland, Mexico, South Africa, Japan, Netherlands, Canada and the United States.

3.1.13 DABS System

DABS (Discrete Address Beacon System) seeks to update the present transponder based system in NAS. The computer portion of the system has as its task the acquisition, processing and management of radar data. Total processing power required is about 3 MIPS, which is handled by 12-15 microprocessor "units."

Each unit consists of two microprocessors strapped together, running on a common clock, and receiving the same inputs. At critical junctures, e.g., before a memory write, the outputs of the two processors are compared by a special hardware device, and if different, the "unit" is considered to have failed and a flag is set. A new unit is then switched in. The system functions with 256K of global memory, and 8K per unit for program storage and scratch pad use. Theoretically, there are no single points of failure which can result in global data contamination. The design goal is 20,000 hours MTBF for the overall system.

To date, only a few of the processing units have been built, and it is not yet known how well the 12 interconnected will function, or if any problems will develop.

This system is similar in type to the MITRE MTI system discussed in Section 3.1.6, and the same remarks in reference to the FSS application apply to it, though of course with respect to reliability DABS is inherently superior and illustrates another approach to fail-safe design.

3.1.14 ARTS III-A

ARTS (Automatic Radar Terminal System) III-A is an enhanced version of ARTS III, designed to provide redundancy, degraded modes of operation and radar tracking.

The system employs two Sensor Reception and Processing units (SRAP) and two Input/Output Processors (IOPs). Each SRAP has 4K bytes of memory, and the IOPSs share a common memory bank of 96K bytes. UNIVAC is developing the system, and the first prototype is scheduled to be operational at Tampa shortly.

This system is not particularly closely related to the FSS application both because of the load (see Sections 3.1.6, 3.1.13), and because the hardware is not standard off-the-shelf commercial minicomputers.

3.1.15 AWANS

AWANS (Automated Weather and NOTAM System) is basically a minicomputer based distributed processing system whose function is to process and display weather data and flight plans, quite similar to the way envisioned for the FSS. It maintains a national aviation weather data base, and permits various methods of retrieval and display of the data. Included are both graphical and live weather data.

The hardware is comprised of three GTE TEMPO II processors, each containing 128K bytes of memory (the maximum for this machine), four cartridge disks, two tape drives, one line printer, two character printers, 18 keyboards and 14" raster scan displays, and three teletypes. The system normally runs with two processors on line, and if one fails, the third is manually switched into the system. Several vendors supplied the hardware, and E-Systems integrated it.

The operating system and applications programs were written in assembly language. The Atlanta system was delivered in June 1975 and is operational. A second system was delivered to Leesburg in February 1977.

Since AWANS was intended as a prototype FSS, it is similar in design and function, but without all the capabilities of the FSS, nor its fail-safe/fail-soft requirements.

3.2 Conclusions from Survey of Existing Distributed Processing Systems

3.2.1 Reasons for Selecting Distributed Processing Architecture

Nearly all users agreed on four reasons for preferring distributed processing: (a) Cost, (b) Reliability, (c) Modularity,

(d) Expansibility. Unfortunately, very little hard data are available on how well distributed systems are living up to their promise of reliability, and for the most part expansibility is theoretical rather than actually demonstrated. As for items (a) and (c), however, there is very little doubt that distributed systems fulfill their promise.

3.2.2 Availability of Hardware

Hardware currently available off-the-shelf, including processors, disk systems, tape drives and other peripherals as well as interprocessor busses, is adequate for the FSS application. There is at least one vendor who seems to have all the hardware needed, including that required for automatic reconfiguration and fail-safe operation, available in a package which is completely off-the-shelf. Most other vendors offer various pieces of the required gear (such as dual ported peripherals, dual busses, special power supplies, etc.), so that is is all available in one form or another, but some development may be required to assemble and integrate it.

3.2.3 Availability of Software

Nearly all vendors offer a real-time multitasking operating system off-the-shelf, which is capable of sustaining fairly sophisticated developmental efforts. There is at least one who

offers a real-time multiprocessing operating system meeting most or all of the FSS requirements, but most sites currently existing either wrote all of their own operating system (e.g., Southern Bell, DCS, C.mmp), or substantially modified existing operating systems (e.g., JETS). Most, however, did not recommend this procedure.

The software to support automatic reconfiguration and fail-safe operation is available off-the-shelf from only one vendor from among those we investigated, and for other equipment would have to be developed in conjunction with the necessary hardware. Several sites did have manual reconfiguration capability.

Higher level language availability is summarized in Section 2.3. The MITRE/FAA study team determined that only one vendor had its COBOL compiler tested by the Navy testing center (Data General), so in most respects compiler performance cannot be guaranteed.

3.2.4 Systems of Comparable Size and Scope

As the MITRE/FAA team concluded, no single operational site or system has been found which approximates the size and workload anticipated for a typical FSS Hub. But when all sites are taken

collectively, most major functions required appear to have been implemented or are being developed.

3.2.5 Recommendations of Current Users

Those who are currently using or developing distributed processing systems made several recommendations which should prove most valuable. First, with regard to hardware they recommended that equipment be purchased which is upward compatible and more powerful than present estimates call for, both with respect to CPU and memory. Nearly all at first underestimated the processing power required for their job. Second, with respect to software, nearly all recommended that higher level languages be used for application programs, and most advocated vendor supplied operating systems, although such obviously do not exist for some of the more exotic architectures, such as DCS. Third, regarding overall system performance, most felt that a comprehensive benchmark was essential to assure that a machine lives up to its specifications. Fourth, most stressed procurement of premium grade hardware and peripherals. The savings realized by buying cheap hardware was used up many times over during later problems with hardware, software development, poor reliability, schedule slips, etc.

4. TWO SAMPLE DESIGNS FOR THE FLIGHT SERVICE STATION HUB

Next, a typical function sharing and a typical load sharing configuration for the FSS system will be described and analyzed. The purpose of this exercise is to show the feasibility of constructing an FSS Hub capable of meeting all requirements of the current version of the Flight Service Station specification using off-the-shelf hardware, and to analyze the expected performance of such a system under peak anticipated load. An evaluation of the results of this section may be found in Section 4.3, and a performance summary in Table 4-28.

There is no implication that these are the only or the best designs for the Flight Service Station, nor that they are the designs recommended by The MITRE Corporation or the Federal Aviation Administration.

This is not a complete analysis and it is based on a number of assumptions. It is intended to substantiate reasonable configurations as examples of possible approaches to the FSS problem.

4.1 Function Sharing Configuration

In this type of configuration, the overall job is split up into functions which are assigned to machines, either dynamically or permanently. Here a permanent assignment will be assumed. A

preliminary design will be illustrated, based on system throughput considerations alone. Then a more powerful system will be shown, and both subjected to a queuing analysis to ascertain their expected level of performance.

4.1.1 Assumptions

To analyze the performance of a function sharing distributed processing system, the following assumptions will be made:

1. FSS functions are broken up into five major categories:
   a. Data Base Maintenance and Update
   b. Flight Plan Processing
   c. Weather Retrieval
   d. Route Processing
   e. I/O Processing, Formatting, and Display; Miscellaneous

2. Data base maintenance is done in the special manner explained below, Section 4.1.3.

3. TANDEM computers will be used for sizing and costing purposes.

4. Speed of TANDEM machines assumed to be that of NAS instruction mix.

5. Overhead is assumed to be 60% of operational software.

6. FSS Hub CPU requirements are as in Appendix B, with modifications as noted.

7. No processing will begin until all entries required have been made.

The effect of relaxing assumption (7) will be discussed in Section 4.3.

The data flow for the three major FSS functions (Route Briefing, Local Briefing, and Flight Plan Filing) are shown in Figures 4-1 through 4-3.

4.1.2 Estimate of Number of Machine Instructions/Second Required by Pilot Briefing Functions

Based on the data in Appendix B, and the function split described in Section 4.1.1, the number of instructions/second required to perform the work in each of the five functional areas can be determined for the 1995 Peak Hour Load.

4.1.2.1 Raw Computing Power Required

The following sections estimate the number of machine instructions/second required by pilot briefing functions.

4.1.2.1.1 Data Base Maintenance and Update

1000 msgs/240 sec* x 10K instr/msg = 42K instr/sec

* Worst case condition for data base update: 1000 = number of SA reporting stations; 240 seconds = time of transmission.

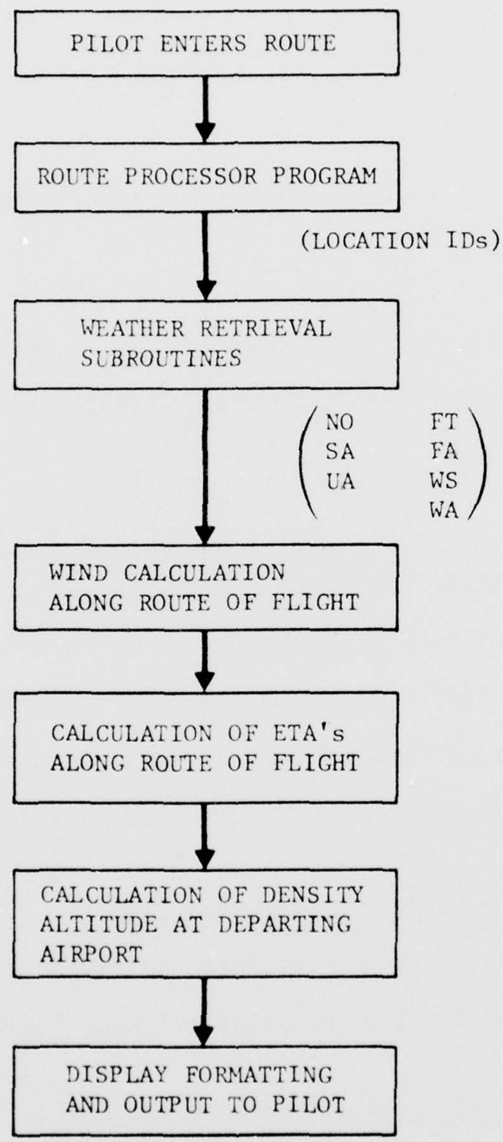


FIGURE 4-1
DATA FLOW: ROUTE ORIENTED BRIEFING

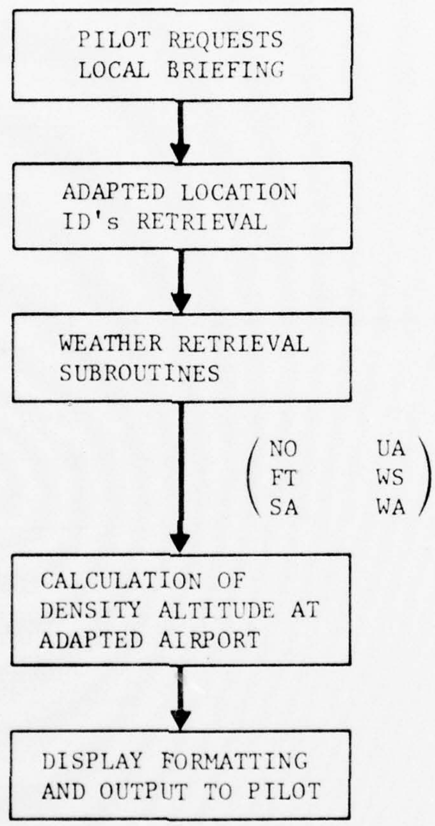


FIGURE 4-2
DATA FLOW: LOCAL WEATHER BRIEFING

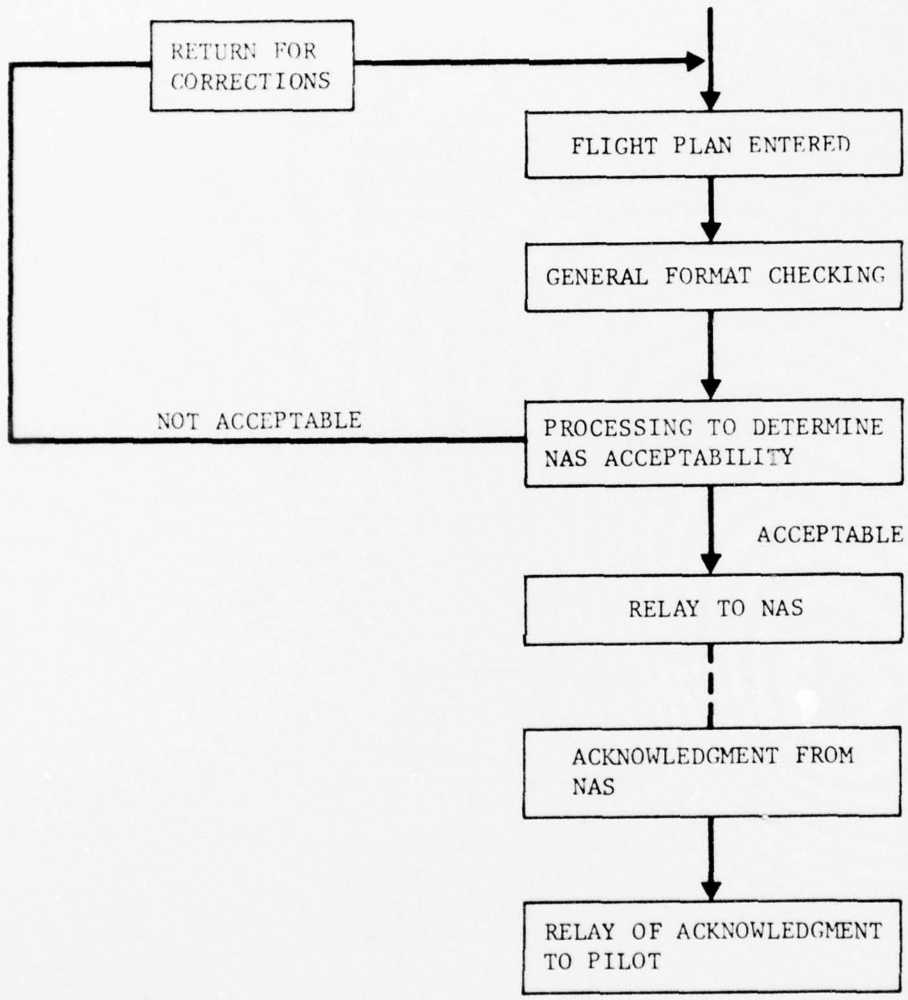


FIGURE 4-3
BASIC DATA FLOW FOR FLIGHT PLAN FILING

AD-A047 930 MITRE CORP MCLEAN VA METREK DIV F/G 9/2
DISTRIBUTED PROCESSING APPLIED TO THE FLIGHT SERVICE STATION MO--ETC(U)
AUG 77 T B FOWLER DOT-FA69NS-162
UNCLASSIFIED MTR-7576 FAA/RD-77-161 NL

2 OF 3
AD
A047930

4.1.2.1.2 Flight Plan Processing and Aircraft Contact Messages

100K instr/IFR flight plan x 299 IFR f.p./hr

x 1 hr/3600 sec = 8300 instr/sec

50K instr/VFR flight plan x 116 VFR f.p./hr

x 1 hr/3600 sec = 1600 instr/sec

20K instr/AC Message x 189 AC Messages/hr

x 1 hr/3600 sec = 1050 instr/sec

11K instr/sec

4.1.2.1.3 Weather Retrieval

Items (b)-(m) from Table B-1 = 680K instr/

briefing x 606 briefings/hr x 1 hr/3600 sec = 116K instr/sec

All items from Table B-2 = 92K instr/briefing

x 307 briefings/hr x 1 hr/3600 sec = 7850 instr/sec

124K instr/sec

4.1.2.1.4 Route Processing

430K instr/route brief x 606 brief/hr x 1 hr/

3600 sec = 72K instr/sec

4.1.2.1.5 I/O Processing, Formatting and Display; Miscellaneous

Formatting and Display:*

Route Oriented Weather Briefings (Specialist):

* Assumes 75% of briefings by Pilot using Direct Access Devices (Interactive Telephone and Terminal), and 25% by all other means.

152 brief/hr x 5000 char/brief x 10 instr/char x 1 hr/3600 sec = 2111 instr/sec

Route Oriented Weather Briefing (Pilot at DUAT):

454 brief/hr x 7540 char/brief x 10 instr/char x 1 hr/3600 sec = 9508 instr/sec

Local Weather Briefing (average for Specialist and Pilot): 307 brief/hr x 1500 char/brief x 10 instr/char x 1 hr/3600 sec = 1279 instr/sec

I/O Processing:

Basic functions (Briefings, Flight Plans, Aircraft Contacts): (415 + 606 + 307 + 189) requests/hr x 10K instr/request x 1 hr/3600 sec = 4214 instr/sec

Miscellaneous (Approx. 2 per each category under 'Basic Functions')

3000 requests/hr x 10K instr/request x 1 hr/3600 sec = 8333 instr/sec

Logging:

4500 messages/hr x 10K instr/message x 1 hr/3600 sec = 12.5K instr/sec

38K instr/sec

4.1.2.2 Computing Power Required Including Overhead and CPU Utilization

Next, the above estimates will be modified to allow for system overhead and CPU utilization.*

4.1.2.2.1 Data Base Maintenance and Update

$$42\text{K}\ \frac{\text{instr}}{\text{sec.}} \times \underbrace{\left(\frac{1}{0.6}\right)}_{\text{CPU Utilization}}\underbrace{(1.6)}_{\text{Executive Overhead}} = 110\text{K instr/sec}$$

4.1.2.2.2 Flight Plan Processing

$$11\text{K instr/sec} \times \left(\frac{1}{0.6}\right)(1.6) = 29\text{K instr/sec}$$

4.1.2.2.3 Weather Retrieval

$$124\text{K instr/sec} \times \left(\frac{1}{0.6}\right)(1.6) = 331\text{K instr/sec}$$

4.1.2.2.4 Route Processing

$$72\text{K instr/sec} \times \left(\frac{1}{0.6}\right)(1.6) = 192\text{K instr/sec}$$

---

* No allowance is made here for expansion capacity, as is done in Section 4.2, because it is assumed that any functional expansion would be done by addition of more processors, and not by increasing the functions assigned to any one machine.

4.1.2.2.5 I/O Processing, Formatting and Display; Miscellaneous

$$38\text{K instr/sec} \times \left(\frac{1}{0.6}\right)(1.6) = 101\text{K instr/sec}$$

4.1.3 Preliminary Configuration and Recommended Configuration

The machines used for this exercise are TANDEM-16 processors, whose speed is estimated as 420K IPS. With this information and that calculated in Section 4.1.2, the number of machines required per Hub may be estimated.

Given the functional split, and considering peak hour throughput, one machine would be adequate for Data Base Maintenance, one for Flight Plan Processing, one for Route Processing, one for Weather Retrieval, and one for Terminal I/O processing, formatting and display.

Because the peripherals are only two port, and one port is reserved for the redundant processor, data base maintenance may be envisioned as working in the following manner in order to offload the weather data base maintenance chores from the weather retrieval processor: one machine receives incoming weather data, prepares it for storage (in blocks ready for transfer to disk), transmits them over the bus to however many processors are engaged in weather retrieval, which then transfer them to disk by DMA channel.

The alternative to this arrangement would be to have each weather retrieval processor (if more than one; see below, p.4-12) do its own data base maintenance and update. In that case, at least 2 weather retrieval machines would be required (331K IPS : 2 = 165K IPS + 110K = 275K IPS). In addition to being a duplication of effort, this technique would reduce response time for weather retrieval and could have another adverse effect: the position of items in the data bases of the machines might not be identical, whereas weather data bases identical in all respect are clearly desirable, especially for portability of disk packs and cartridges, for copying of the data base, etc. Most operating systems have a provision for reserving tracks on disks which the operating system's file handling programs will not touch, and this feature could clearly be used to advantage in this case. The master copy of the weather data base would remain with the communications handling processor, and could be transmitted to as many weather retrieval machines as necessary - a potentially important consideration for expandability.

All processors are assumed to have 384K bytes of memory. This would allow for multiple copies of operational programs to be stored in memory, so as to speed execution when multitasking is required.

Each processor is assigned two moving head disks with one spare for redundancy. Each machine will also have one tape drive, but because the tape drive is not used for on-line processing, it is not duplicated. It may, however, be duplicated on any machines used for logging.

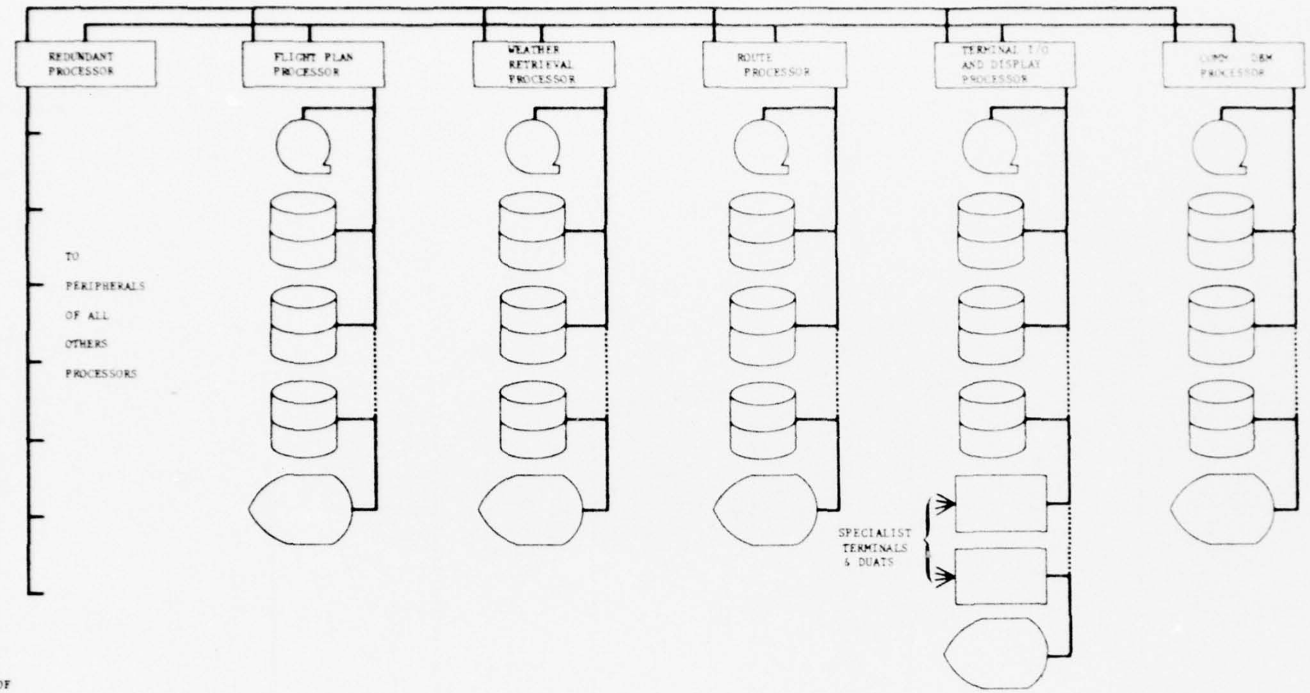
A tentative configuration based on the foregoing considerations is shown in Figure 4-4, with the machine utilization indicated.

Significant system improvement may be realized by reducing the utilization in processors heavily loaded (e.g., the weather retrieval processor) or the processors handling tasks requiring large amounts of serial processing (e.g., the route processor). Assuming that these two are in fact duplicated, a second configuration is obtained, and shown in Figure 4-5.

The expected performance of both of these configurations is analyzed in Section 4.1.4, the first referred to as the "minimal configuration", the second as the "recommended configuration".

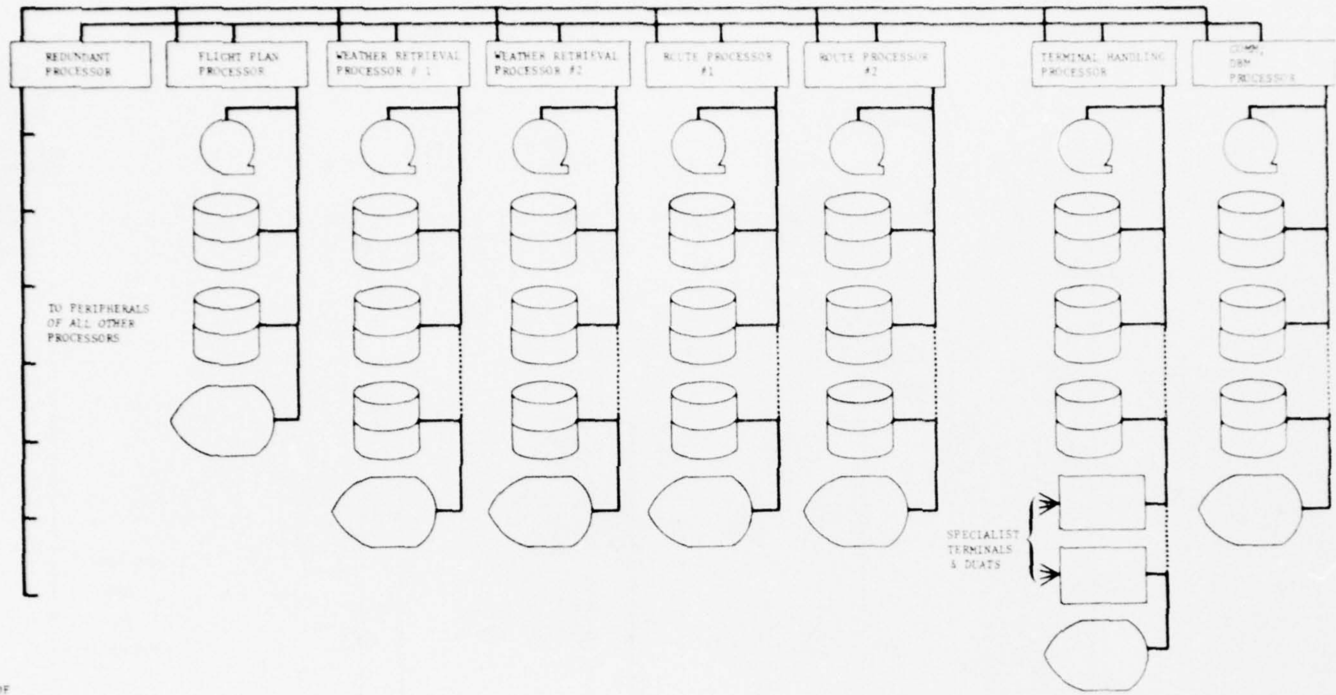
4.1.4 Queueing Considerations For A Function Sharing FSS System

In the following sections a queueing analysis is performed in order to estimate the response time of the system for the major functions.



	Redundant Processor	Flight Plan Processor	Weather Retrieval Processor	Route Processor	Terminal I/O and Display Processor	Comm O&M Processor
BYTES OF SEMI-CONDUCTOR MEMORY	384K	384K	384K	384K	384K	384K
AVERAGE INSTRUCTIONS EXECUTED/SEC		29K IPS	331K IPS	192K IPS	101K IPS	110K IPS
MACHINE UTILIZATION: (EXCLUDING CPU UTILIZATION FACTOR)		.04	.47	.27	.14	.16

FIGURE 4-4
TENTATIVE FUNCTION SHARING FSS HUB
CONFIGURATION EMPLOYING TANDEM COMPUTERS



	Redundant Processor	Flight Plan Processor	Weather Retrieval Processor # 1	Weather Retrieval Processor #2	Route Processor #1	Route Processor #2	Terminal Handling Processor	Comm, DBM Processor
BYTES OF SEMI-CONDUCTOR MEMORY	384K	384K	384K	384K	384K	384K	384K	384K
INSTRUCTIONS EXECUTED/SEC (PEAK HOUR)		29K	166K	166K	96K	96K	101K	110K
MACHINE UTILIZATION (EXCLUDING CPU UTILIZATION FACTOR		.04	.24	.24	.14	.14	.14	.15

FIGURE 4-5
RECOMMENDED FUNCTION SHARING FSS HUB CONFIGURATION FOR IMPROVED RESPONSE TIME USING TANDEM COMPUTERS

4.1.4.1 Identification of Time-Critical Areas

In the FSS System, response time is most likely to be a problem in those functions which involve a series of tasks that are basically serial in nature, i.e. permit very little overlapping, require relatively large amounts of processing time, and large numbers of disk accesses. All major functions will be considered in this paper, but the worst case is the route-oriented briefing. In Figure 4-6 is a schematic diagram of data flow in a route-oriented briefing, with the important queuing points identified, together with the type of queing model assumed for each. Usually the most general model was chosen (M/G/1)*, since service times will depend on hardware configuration and especially on the actual software used, about which, of course, little information is available. But a model had to be chosen and certain assumptions about software made. The hardware was assumed to be TANDEM, and the software characteristics those of the application programs running on the MITRE PDP 11/70 experimental PSBT System. In many cases estimates of distributions had to be made, but where possible these represent worst-case or near worst-case conditions, so that the performance estimates obtained should be conservative.

---

* The usual abbreviated notation for queuing models is employed for this report. In that notation, for example A/B/C, A represents the interarrival time distribution, which is generally assumed to be exponential and thus corresponds to a Poisson distribution for arrival rate; B is the service time distribution, usually either indicated as "G" for general, or E_k for Erlang-k distribution; C is the number of servers in the system.

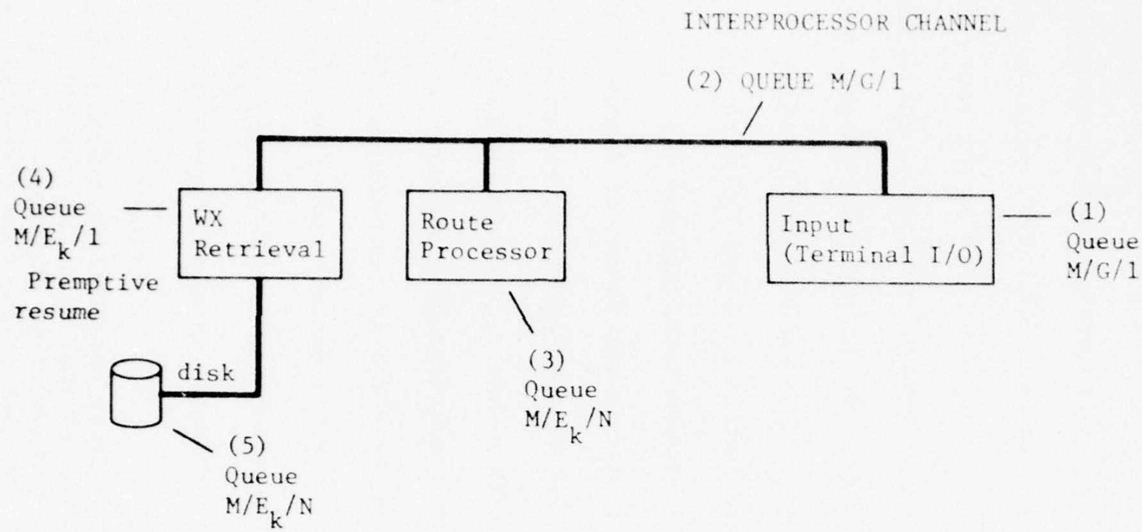


FIGURE 4-6
QUEUEING POINTS IN A FUNCTION SHARING DISTRIBUTED PROCESSING CONFIGURATION

However, due to relatively low processor utilization, the model is not particularly sensitive to those assumptions.

4.1.4.2 Queuing Calculations

In this section the data on which response time estimates will be based is gathered and the necessary calculations made for each of the areas indicated in Figure 4-6.

4.1.4.2.1 Input (Terminal I/O processing) - 1995 Demand

A. Load/hour

454 PSBT briefs @ 7500 Char/brief x 10 instr/char (terminal output)	= 34.05 M instr/hr
Input processing @ 10K instructions each	= 4.54 M
152 Specialist briefs @ 5000 char/brief x 10 instr/char (terminal output)	= 7.60 M
Input Processing @ 10K instructions each	= 1.52 M
303 Local briefs @ 1500 char/brief x 10 instr/char	= 4.55 M
Input Processing @ 10K instructions each	= 3.03 M
189 Aircraft Contracts @ 20K instr each	= 3.78 M
415 Flight Plans @ 10K instr each	= 4.15 M
4500 Logging messages @ 10K instr each	= 45.00 M
3,000 Miscel. messages @ 10K instr each	= 30.00 M
	138.2 M instr/hr
	= 38K instr/sec

B. Queuing parameters - M/G/1 queuing model. This particular queuing model was chosen for the input processor because it is very general and makes minimal assumptions about the nature of the system modeled. All tasks are assumed to be of equal priority.

For this model, expected queuing time is given by the Pollaczek-Khintchine equations*, specifically,

$$E(q) = \frac{\lambda(\sigma_s^2 + \bar{s}^2)}{2\,(1-\rho)} = \frac{\rho\,(\bar{s} + \sigma_s^2/\bar{s})}{2\,(1-\rho)}$$

$E(w)$ = total waiting time in system = $S_i + E(q)$

where ρ = utilization

s = mean service time

σ_s^2 = variance of service time

S_i = service time for task i

Based on the above data, 1995 peak hour utilization for the input processor is given by

$$= \frac{\overset{\text{tasks}}{38\text{K instr/sec}}}{\underset{\text{equivalent machine speed}}{262\text{K instr/sec}}} = .15$$

---

* A.D. Allen, Elements of Queuing Theory for the System Design. IBM Systems Journal, No. 2, 1975, p. 185

Utilization can be calculated for other cases of interest, using the data in Appendices A and B. The two parameters $\bar{s}$ and σ_s^2 are the mean and variance of the service time, and can be determined by calculating the first and second moments of service time using the standard formulae:

$$E(s) = \bar{s} = \sum_i P_i s_i$$

$$E(s^2) = \overline{s^2} = \sum P_i s_i^2$$

$$\sigma_s^2 = \overline{s^2} - (\bar{s})^2$$

where P_i = relative frequency of task i

Breaking down the tasks performed by the input processor, and assuming that terminal output is done in pages of about 500 characters each, the above quantities may be determined based on the data in Appendices A and B, and the results are summarized in Table 4-1.

TABLE 4-1

DATA FOR ESTIMATING RESPONSE TIME VARIANCE

Task	Number/hr.	P_i	s_i	$P_i\ s_i (= \bar{s})$	$P_i s_i^2$
1. PSBT pages	6810	.373	.019	7.08×10^{-3}	1.35×10^{-4}
2. PSBT input proc.	454	.025	.038	9.44×10^{-4}	3.59×10^{-5}
3. Specialist pages	1520	.083	.019	1.58×10^{-3}	3.00×10^{-5}
4. Specialist input proc.	152	.008	.038	3.16×10^{-4}	1.20×10^{-5}
5. Local brief pages	921	.050	.019	9.58×10^{-4}	1.82×10^{-5}
6. Local brief input	307	.017	.038	6.39×10^{-4}	2.43×10^{-5}
7. Aircraft contacts	189	.010	.076	7.86×10^{-4}	5.98×10^{-5}
8. Flight Plans	415	.023	.057	1.29×10^{-3}	7.38×10^{-5}
9. Logging	4500	.246	.038	9.36×10^{-3}	3.56×10^{-4}
10. Misc. messages	3000	.164	.038	6.24×10^{-2}	2.37×10^{-4}
Totals	18268			2.92×10^{-2}	9.81×10^{-4}

Hence,

$$\sigma_s^2 = \overline{s^2} - (\bar{s})^2 = 9.81 \times 10^{-4} - (2.92 \times 10^{-2})^2 = 1.29 \times 10^{-4}$$

This is strictly speaking accurate only for 1995 peak hour load. However, the relative proportions of the different types of tasks remains nearly constant for different years and loads (see Appendix A), so these values for the parameters will be assumed valid for the four system load conditions in the cases considered in this report.

Hence, for each of the four cases of interest, ρ may be calculated and using the above parameter values, E(q) determined. The results are tabulated below:

TABLE 4-2

E(q) FOR FUNCTION SHARING CONFIGURATION

	RECOMMENDED CONFIGURATION			MINIMAL CONFIGURATION
ITEM	1985 PEAK HR.	1995 PEAK HR.	1995 AVERAGE HR.	1995 PEAK HR.
ρ	.10	.15	.07	.15
E(q)	.002	.003	.001	.003

where ρ = utilization, E(q) = expected queuing time.

To determine E(w) for any task handled by the input processor, it is only necessary to add the correct E(q) from the above table to the calculated service time.

4.1.4.2.2 Interprocessor Channel Utilization - 1995 Demand

Capacity (per bus) = 13M bits/sec = 1.625M bytes/sec

Peak hour bus traffic (bytes) is estimated as follows:

1. From weather retrieval to input processor:

454 PSBT briefs @ 7500 char/brief = 3.41M bytes/hr

2. To route processor from input processor:

454 PSBT briefs @ 500 char/brief = 0.23M bytes/hr

3. From weather retrieval to input processor:

152 Specialist briefs @ 5000 char/brief = .76M bytes/hr

4. To route processor from input processor:

152 Specialist briefs @ 500 char/brief = .08M bytes/hr

5. Between processors:

monitor messages @ 250 char/sec* = .90M bytes/hr

6. To weather retrieval from route processor:

606 route briefs @ 1500 char/briefs = .91M bytes/hr

7. From weather retrieval to input processor:

303 local briefs @ 1500 char/briefs = .45M bytes/hr

---

* Assumes about 30 character/message, 8 messages/sec. (Both estimates.)

8. To weather retrieval from input processor:

303 local briefs @ 500 char/brief = .15M bytes/hr

9. To/from flight plan processor:

415 flight plans @ 1000 char each = .42M bytes/hr

10. To/from comm. processor:

189 aircraft contact messages @ 1000 char. each = .19M bytes/hr

11. 3,000 miscellaneous messages @ 500 char. each = 1.5M bytes/hr

12. Weather data update = .47M bytes/hr

9.4M bytes/hr

= 2611 bytes/second

Bus utilization may be calculated:

$$\rho = \frac{2611 \text{ bytes/sec}}{1.625 \times 10^6 \text{ bytes/sec}} = .0016$$

With bus utilization so low, essentially regardless of the queueing model assumed it will have no effect on waiting time, and so will be ignored in the remainder of this study.

4.1.4.2.3 Route Processor

Studies on the route processing program now running on the experimental PSBT system at MITRE have shown an approximately linear relationship between route distance and number of instructions executed (instruction = 348K + 180 x distance).* The minimum number of instructions is about 348K, and the maximum about 750K. Added to this are 11 disc accesses, which may be assumed to take 52.5/msec each, for a total of about .6 seconds. At 262K instr/sec, this yields for the total route processing time:

$$\frac{348K}{262K} + .6 \text{ sec} \leq t \leq \frac{750K}{262K} + .6 \text{ sec}$$

$$1.928 \text{ sec} \leq t \leq 3.463 \text{ sec}$$

Based on the average route distance for a route oriented briefing of 480nm,** the average time for a route processing task will be:

$$\frac{348K + (180)(480)}{262K} + .6 = 2.26 \text{ sec.}$$

No exact data on the distribution of route distances are available, but FSS specialists estimate that 85% are less than 500nm,

* See Appendix B for details.

** See Appendix B, Section 2.5.

so if we have a distribution which closely restricts the briefings to the average, that will be a reasonable estimate for the purposes of this queueing analysis.

The distribution chosen to represent the distribution of route processing service times is an Erlang-10 distribution, which is a conservative assumption and so will put a reasonable upper limit on queueing time and waiting time. The route processor is assumed to work on only one task at a time, and processing is lumped with disk I/O here for the purpose of calculation. The route processor may be modeled by an $M/E_k/1$ queueing system, for which we have the following parameters:

For Erlang-10 distribution of service time:

$$E(x) = \alpha = 2.26 \text{ sec}$$

$$E(x^2) = \frac{k+1}{k} [E(s)]^2 = \frac{11}{10} [E(x)]^2 = 1.1\alpha^2 = 5.618$$

The following cases of interest may be distinguished:

I 1985 Peak hour, 2 route processors

II 1995 Peak hour, 2 route processors

III 1995 Average hour, 2 route processors

IV 1995 Peak hour, 1 route processor.

For Case IV, the mean waiting time (= service time + queueing time) can be calculated, using the following formulae for a single server queue:

α = mean service time = $\bar{s}$

λ = mean arrival rate per second (= number/hr $\div$ 3600)

$\rho = \lambda\alpha$ = system utilization

$E(x^2) = 1.1\alpha^2$

$Wq = \dfrac{\lambda E(x^2)}{2(1-\rho)}$ = mean waiting time in queue

$E(w) = Wq + \alpha$ = mean waiting time

For cases I, II and III, the formulae for a multiserver queueing system* must be employed. This is because in each of these cases there are assumed to be two processors available for route processing, and the system will assign tasks to them as a function of their workload. The route processors, in other words, form a two-server system. With the same parameters as before, we define

$$Z = \left(\frac{\sum_{n=0}^{m-1} \frac{(m\rho)^n}{n!}}{\sum_{n=0}^{m} \frac{(m\rho)^n}{n!}} \right)$$

where m = number of servers. Then the probability that all are busy is B, given by

$$B = \frac{1 - z}{1 - \rho z}$$

* Systems Analysis for Data Transmission, James Martin, Prentice Hall: Englewood Cliffs, 1972, p. 451-461.

Mean waiting time in queue is

$$W_q = \frac{B}{2M} \left[\frac{\bar{s} + \sigma_s^{2}/\bar{s}}{1 - \rho} \right] = \frac{B}{2M} \; \frac{(1.1)(\bar{s})}{(1 - \rho)}$$

and

$$E(w) = W_q + \alpha$$

The results of carrying out the above calculations for the four cases are summarized in Table 4-3.

TABLE 4-3

ROUTE PROCESSING SERVICE TIMES FOR FUNCTION SHARING CONFIGURATION

	CASE I	II	III	IV
Mean number/hour/ processor.	207	303	152	606
α	2.26	2.26	2.26	2.26
λ	.058	.084	.042	.168
ρ	.130	.190	.095	.380
$E(x^2)$	5.618	5.618	5.618	5.618
B	.0299	.0608	.0166	-
W_q	.0214	.0467	.0114	.763
E(w)	2.281	2.307	2.271	3.023

Due to the relatively low number of disk accesses per hour, for the recommended configuration we have,

$$\frac{303 \text{ briefs}}{\text{machine}} \times \frac{11 \text{ accesses}}{\text{brief}} = 3333 \text{ accesses/hr} = .93/\text{sec}$$

Hence, allowance will be made for only two disks per route processor in the recommended configuration, and three in the minimal configuration.

4.1.4.2.4 Weather Retrieval

Weather retrieval for route oriented and local briefings is envisioned as working in the following manner. When a request for one of these two types is input, the system assigns highest priority to retrieving and displaying the first page of the weather, since the rest of the pages can be retrieved while the specialist or pilot is reading the first page. It is estimated that either will take at least five seconds to scan the page. The remaining weather data is then retrieved and sent to the terminal I/O processor to be ready for display when the requestor is ready.

One further matter must be analyzed regarding the first page of displayed weather data. According to the FSS specification, the specialist or pilot has three choices as the first item in his briefing: (1) graphics, (2) FA's, (3) alphanumeric briefing. There are of course no statistics available on how often each of these will be selected, but a reasonable assumption is: graphics, 60%, FA's, 20%; and alphanumeric briefing (SA's first), 20%. Using the data in Appendix B, and assuming two disk accesses for retrieval of an AFOS chart, this yields a weighted average number of instructions of about 16K, and between four and five

disk accesses. Here, 16K instructions and five disk accesses will be assumed.

In view of the numerous functions the weather retrieval processor must perform, some of which, as indicated above, are quite lengthy, optimum performance can be realized only with a priority queueing scheme.

Here the number of tasks is divided into nine categories, and an M/G/1 preemptive-resume queueing model assumed. The categories are depicted in Figure 4-7.

This scheme has been designed to give highest priority to retrieval and output of the first page of a briefing, since this time will have most effect on overall system response time. As explained above, it is assumed that a specialist or pilot will take at least several seconds to scan the first page, so retrieval and output of the remainder of the items in the briefing is relegated to lowest priority.

As before, worst-case or near worst-case assumptions were made concerning service time distribution, so that the calculated waiting time estimates should be conservative. Data for the number of instructions for items in each category were taken from Tables B-1 and B-2.

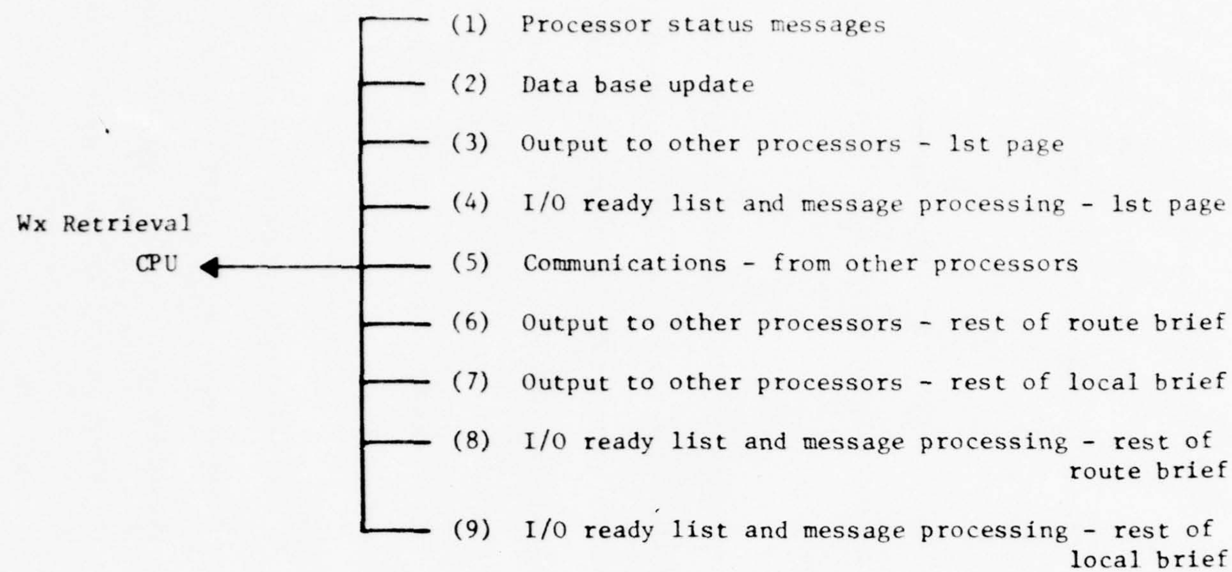


FIGURE 4-7
CATEGORIES IN PREEMPTIVE–RESUME WEATHER RETRIEVAL QUEUEING SCHEME

In order to evaluate the mean waiting time for items in each of these categories, it is necessary to know λ (mean arrival rate), $\alpha^{(1)}$ (mean service time), and $\alpha^{(2)}$ (second moment of service time) for each. The last two items, of course, remain unchanged regardless of λ, i.e., regardless of the number of briefs/processor given per hour. The usual four cases are distinguished below: (1) two processors, 1985 peak hour load; (II) two processors, 1995 peak hour load (which corresponds to one processor average hour load); (III) two processors, 1995 average hour load; and (IV) one processor, 1995 peak hour load. Relevant data for the cases are summarized below (demand figures from Appendix A).

Case I: two processors, 1985 peak hour load, per processor demand

Route Briefs: 207

Local briefs: 101

Case II: two processors, 1995 peak hour load, per processor demand

Route briefs: 303

Local briefs: 154

Case III: two processors, 1995 average hour load, per processor demand

Route briefs: 152

Local briefs: 77

Case IV: one processor, 1995 peak hour load

Route briefs: 606

Local briefs: 307

In Table 4-4, the nine individual categories are given in the left column, together with their respective $\alpha^{(1)}$ and $\alpha^{(2)}$; to the right, the corresponding value of λ is given for each of the four cases.

A preemptive-resume queuing model is assumed here, for which the relevant formuale are given below:*

$$U_j = \text{traffic intensity} = \sum_{i=1}^{j} \lambda_i \, E(s_i) = \sum_{i=1}^{j} \lambda_i \, \alpha_i^{(1)}$$

$$E(w_j) = \text{mean waiting time} = \frac{1}{1 - U_{j-1}} \left[\alpha_j^{(1)} + \frac{\sum_{i=1}^{j} \lambda_i \, \alpha_i^{(2)}}{2(1 - U_j)} \right]$$

Results of the queueing calculations are given in Table 4-5 for the nine categories in each of the four cases. Also given is the service time for each item, so that a direct comparison may be made with the waiting time calculated. All numbers are given to three significant figures.

* Allen, op. cit., p. 186-187.

TABLE 4-4

VALUES OF λ (MEAN ARRIVAL RATE) FOR DIFFERENT LOADS IN PREEMPTIVE RESUME WEATHER RETRIEVAL QUEUEING SCHEME

	CASE			
CATEGORY	I	II	III	IV
1. Processor status messages @ 75 instr ea. Service time assumed constant $\alpha^{(1)} = .0003$ sec $\alpha_1^{(2)} = 8 \times 10^{-8} \text{sec}^2$	28800/hr = 8/sec = λ_1	28800/hr = 8/sec = λ_1	28800/hr = 8/sec = λ_1	21600/hr = 6/sec = λ_1
2. Data base update* @ 500 instr each E_3 service time distribution $\alpha^{(1)} = .0019$ sec $\alpha^{(2)} = 4.9 \times 10^{-6}$ sec	90/hr = .025/sec = λ_2	90/hr = .025/sec = λ_2	90/hr = .025/sec = λ_2	90/hr = .025/sec= λ_2
3. Output to other processors, 1st page @ 5000 instr each** E_3 service time distribution $\alpha_3^{(1)} = .01908$ sec $\alpha_3^{(2)} = .00049 \text{ sec}^2$	207 + 101 = 308/hr = .0856/sec = λ_3	303 + 154 = 457/hr = .1269/sec = λ_3	152 + 77 = 229/hr = .06361/sec = λ_3	606 + 307/hr = 913/hr = .2536/sec = λ_3

* This assumes the special method of Weather data base management described in Section 4.1.3. However, if each machine performed all the base management functions, the effect would be to cause response times for items 2-7 to increase by about 25-30%, and items 8 and 9 by somewhat more for Case II. With the method chosen here, 90 updates per hour, or about one every 40 secons is assumed. But this frequency can be increased by a factor of three to four with essentially no impact on response time.

** Allows ten instructions executed per character.

TABLE 4-4
(CONTD)

CATEGORY	I	II	III	IV
4. I/O ready list and message processing - 1st page @ 16000 instr each E_3 service time distribution $\alpha_4^{(1)}$ = .0612 sec $\alpha_4^{(2)}$ = .00497 sec^2	308/hr = .0856/sec = λ_4	457/hr = .1269/sec = λ_4	229/hr = .06361/sec = λ_4	913/hr = .2536/sec = λ_4
5. Communications - from other processors @ 10K instr each E_2 service time distribution $\alpha_5^{(1)}$ = .0382 sec $\alpha_5^{(2)}$ = .00222 sec^2	308/hr = .0856/sec = λ_5	457/hr = .1269/sec = λ_5	229/hr = .06361/sec = λ_5	913/hr = .2536/sec = λ_5
6. Output to other processors - rest of route brief* @ 64K instr each E_3 service time distribution $\alpha_6^{(1)}$ = .2443 sec $\alpha_6^{(2)}$ = .0796 sec^2	207/hr = .0575/sec = λ_6	303/hr = .0842/sec = λ_6	152/hr = .0422/sec = λ_6	606/hr = .1683/sec = λ_6
7. Output to other processors - rest of local brief @ 10K instr each E_3 service time distribution $\alpha_7^{(1)}$ = .0382 sec $\alpha_7^{(2)}$ = .00194 sec^2	101/hr = .0281/sec = λ_7	154/hr = .04278/sec = λ_7	77/hr = .02139/sec = λ_7	307/hr = .08528/sec = λ_7

* Assumes weighted average of 12.75 additional pages based on 25% - 75% Specialist - Pilot self-brief split.

TABLE 4-4
(CONTD)

CATEGORY	I	II	III	IV
8. I/O ready list and message processing - rest of route brief @ 671K instr each E_2 service time distribution $\alpha_8^{(1)} = 2.56$ sec $\alpha_8^{(2)} = 9.84\ sec^2$	207/hr = .0575/sec = λ_8	303/hr = .0842/sec = λ_8	152/hr = .0422/sec = λ_8	606/hr = .1683/sec = λ_8
9. I/O ready list and message processing - rest of local brief @ 76K instr each E_3 service time distribution $\alpha_9^{(1)} = .290$ sec $\alpha_9^{(2)} = .112\ sec^2$	101/hr = .0281/sec = λ_9	154/hr = .04278/sec = λ_9	77/hr = .02139/sec = λ_9	307/hr = .08528/sec = λ_9

TABLE 4-5

MEAN CPU WAITING TIMES FOR FUNCTION SHARING CONFIGURATIONS (SECONDS)

CATEGORY	SERVICE TIME	CASE I	CASE II	CASE III	CASE IV
1. STATUS MESSAGES	.000300	.000300	.000300	.000300	.000300
2. D.B. UPDATE	.00190	.00190	.00190	.00190	.00190
3. 1ST PAGE OUTPUT	.0191	.0192	.0192	.0191	.0192
4. 1ST PAGE PROCESSING	.0612	.0617	.0619	.0616	.0623
5. COMMUNICATIONS	.0382	.0389	.0392	.0387	.0401
6. OUTPUT - ROUTE BRIEF	.244	.250	.253	.249	.261
7. OUTPUT - LOCAL BRIEF	.0382	.0420	.0439	.0410	.0503
8. REST OF ROUTE BRIEF	2.56	2.99	3.25	2.86	4.61
9. REST OF LOCAL BRIEF	.290	.777	1.16	.613	4.23

A few comments on this Table are in order. As Items 8 and 9 in the last column show, the design represented by Case IV (1 weather retrieval processor) begins to show signs of performance degradation under expected peak hour loading. However, that degradation is relatively minor, and indicates that with the recommended design and equivalent load (Case II), there is a considerable margin of safety in the form of reserve capacity.

4.1.4.2.5 Disk Queueing

To analyze the queues associated with disk accesses, the same four cases as above will be considered, and for each case, the expected queueing time assuming 1, 2, 3 or 4 disks share the load will be determined.

Based on Tables B-1 and B-2 from Appendix B, the total number of disk accesses per hour for each case can be determined by summing the number for each briefing type, and multiplying by the number of briefings of each type from Appendix A. In addition, an average of 360 accesses per hour per disk are assumed for data base maintenance, but as will become apparent, that number could be several times larger with no appreciable effect on queueing time.* A sample calculation, that for Case IV, is as follows:

* See Reference 1 in Appendix C; 360 one sector transfers/hour corresponds to about 184K bytes/hour; or 40% of estimated peak hour update traffic.

1995 peak hour route briefs: 606

Disk accesses/brief (weather portion): 82 Product: 49692

1995 peak hour local briefs: 307

Disk Accesses/brief (weather portion): 30 Product: 9210

Total disk accesses: 58902 + 360(accesses for d.b.m.) = 59262

The expected load per processor (excluding data base maintenance) is shown in Table 4-6 below.

TABLE 4-6

DISK ACCESSES PER HOUR FOR WEATHER RETRIEVAL UNDER DIFFERENT LOADS

	I	II	III	IV
NUMBER OF ACCESSES	20004	29466	14733	58902

The following assumptions concerning the disks will be made (performance slightly below that of IBM 3330 disk drive):

1. All are identical, moving-head type
2. Rotational speed is 2400 RPM
3. Average seek time is 37.5 msec
4. Transfer rate is 200K bytes/sec
5. There are 10 sectors per track
6. Transfers are one sector which is 512 bytes (1-2% of transfers may be up to one track, but this will have essentially no effect on mean response or queueing time)

7. There is no waiting time for the DMA

An average transfer then will take

37.5 msec (seek) + 12.5 msec (latency) + 2.5 msec

(read 1 sector) = 52.5 msec (= α)

For the purposes of this analysis, all three parts of the disk operation will be lumped together and service time variance will be assumed to be $\bar{s}^2$ (i.e., exponential distribution) which is certainly a worst case assumption. In case a disk transfer is blocked due to a busy channel, the disk must wait another 25 msec before reattempting its transfer. This need not be considered, however, if channel utilization is sufficiently low. For one sector transfers, DMA utilization is summarized in Table 4-7 for the four cases, assuming 1-4 disks. A sample calculation is as follows:

Case IV, 1 disk = 58902 + 360 = 59262 accesses/hr

x .0025 sec/read x hr/3600 sec = .04115

TABLE 4-7

DMA UTILIZATION FOR WEATHER RETRIEVAL PROCESSORS

NUMBER OF DISKS	CASE I	CASE II	CASE III	CASE IV
1	.01414	.02071	.01048	.04115
2	.00720	.01048	.00537	.02070
3	.00488	.00707	.00366	.01388
4	.00372	.00537	.00281	.01048

As is apparent from the table, utilization is well below 2% in all cases likely to be of interest, and below 1% in most, so we shall continue, assuming a multiserver queueing system. The equations used are the same as those employed in the route processor estimation, Cases I, II and III.

$$W_q = \frac{B}{2m} \quad \frac{s + \sigma_s^2 /s}{1 - \rho} = \frac{Bs}{m(1 - \rho)} \quad \text{for } \sigma_s = s,$$

and

$$E(w) = W_q + \alpha$$

Table 4-8 summarizes important data for each of the four cases selected.

Based on these data, the number of disks per weather retrieval processor will be assumed to be three for a two processor system, and four for a one processor system, since this will provide for extremely short queues and a wide margin of safety in case of higher loading due to a failed disk or higher than anticipated demand.

4.1.4.3 Estimation of Total Response Time

The response time for route and local briefings for two configurations under different loading conditions can now be estimated. Response time is considered to be the interval between depression of the ENTER key after the last item of data is input,

TABLE 4-8

MEAN DISK ACCESS WAITING TIMES FOR DIFFERENT LOAD CONDITIONS AND DIFFERENT NUMBERS OF DISKS/PROCESSOR

NO. OF DISKS(m)	E(w)	W_q	B	ρ	AVG. ACCESSES/ DISK/HOUR
CASE I: 2 WX RETRIEVAL PROCESSORS, 1985 PEAK LOAD					
1	.0747	.0222	.2970	.2970	20364
2	.0537	.0012	.0397	.1511	10362
3	.0526	7.7×10^{-5}	.0040	.1025	7028
4	.0525	4.5×10^{-6}	.0003	.0782	5361
CASE II: 2 WX RETRIEVAL PROCESSORS, 1995 PEAK LOAD					
1	.0929	.0404	.4350	.4350	29826
2	.0546	.0021	.0794	.2201	15093
3	.0527	.0002	.0111	.1485	10182
4	.0525	1.8×10^{-5}	.0012	.1127	7727
CASE III: 2 WX RETRIEVAL PROCESSORS, 1995 AVERAGE LOAD					
1	.0673	.0148	.2201	.2201	15093
2	.0532	.0007	.0228	.1127	7727
3	.0525	3.3×10^{-5}	.0018	.0769	5271
4	.0525	1.5×10^{-6}	.0001	.0590	4043
CASE IV: 1 WX RETRIEVAL PROCESSOR, 1995 PEAK LOAD					
1	.3867	.3342	.8642	.8642	59262
2	.0647	.0122	.2635	.4347	29811
3	.0541	.0016	.0652	.2916	19994
4	.0527	.0002	.0133	.2200	15086

ρ = DISK UTILIZATION
B = MULTIPLE SERVER FACTOR
W_q = QUEUEING TIME
E(w) = EXPECTED SERVICE TIME

until the first character of the response appears on the entering terminal. The two configurations considered are: (1) the recommended configuration, consisting of two weather retrieval processors and two route processors, and (2) a minimal configuration, consisting of one weather retrieval and one route processor. The following sections explain how each estimate is made.

4.1.4.3.1 Route Briefing, First Page

1. Input request - assumed to require processing of 10K machine instructions, the time for which is estimated by:

$$\underbrace{10K}_{\text{input request}} \times \underbrace{1.6}_{\text{system overhead}} \;/\; \underbrace{420K}_{\text{machine speed (IPS)}} + \underbrace{E(q)}_{\text{Queueing delay from Sec. 4.1.4.2.1}}$$

2. Route Processing - times from Table 4-3 used directly.

3. Weather Retrieval Processing: CPU, first page - items (3), (4) and (5) from Table 4-5.

4. Disk accesses - first page, average of five assumed, average time for each from Table 4-8, overlapping of seeks and reads assumed to reduce total time by a factor of two.

5. Output - first page, requiring 5K machine instructions for formatting, protocol, etc., the total time for which is

$$5K \times 1.6 \; / \; 420K \; + \; E(q)$$

Output

System Overhead

Machine Speed (IPS)

Queueing delay from Sec. 4.1.4.2.1

6. Interprocessor communications - a minimum of four would be required for output of the first page, but allowance will be made for six in case of problems. TANDEM has stated that each such transaction requires about seven msec.

4.1.4.3.2 Local Briefing - First Page

Same as above, except item (2) deleted, and only four interprocessor messages assumed.

4.1.4.3.3 Route Briefing - Remainder

1. Weather Retrieval Processing: CPU, all other pages - items (6) and (8) from Table 4-5.

2. Weather Retrieval Processing: Disk accesses, all other pages - average of 77 assumed, average time for each from Table 4-8, overlapping of seeks, etc., assumed to reduce total time by a factor of two.

3. Output - single page, same as (5) under "Route Briefing, first page," above.

4. Interprocessor Communication - maximum of two required.

4.1.4.3.4 Local Briefing - Remainder

1. Weather Retrieval Processing: CPU, all other pages - items (7) and (9) from Table 4-5.

2. Weather Retrieval Processing: Disk access, all other pages - average of 25, average time for each from Table 4-8, overlapping of seeks, etc., assumed to reduce total time by a factor of two.

3. and 4. - Same as Route Briefing.

4.1.4.3.5 Route Briefing - Calculations and Final Results

1. First Page - Total response time for output of first page of route oriented briefing is given in Table 4-9.

2. Remainder of Pages - Total time until rest of pages in route briefing are available for display is summarized in Table 4-10.

TABLE 4-9

MEAN RESPONSE TIMES (SECONDS) FOR FIRST PAGE OF ROUTE ORIENTED BRIEFING IN FUNCTION SHARING CONFIGURATION

ITEM	RECOMMENDED CONFIGURATION CASE I 1985 PEAK HOUR	CASE II 1995 PEAK HOUR	CASE III 1995 AVG. HOUR	MINIMAL CONFIGURATION CASE IV 1995 PEAK HOUR
1. INPUT REQUEST	.040	.041	.039	.041
2. ROUTE PROCESSING	2.281	2.307	2.271	3.023
3. WX RETRIEVAL	.120	.120	.119	.122
4. DISK I/O	.132	.132	.131	.132
5. OUTPUT	.021	.022	.020	.022
6. INTERPROCESSOR	.042	.042	.042	.042
TOTALS (SECONDS)	2.636	2.664	2.622	3.382

TABLE 4-10

MEAN TIMES FOR RETRIEVAL OF REMAINDER OF DATA FOR ROUTE ORIENTED BRIEFING IN FUNCTION SHARING CONFIGURATION

ITEM	RECOMMENDED CONFIGURATION CASE I 1985 PEAK HOUR	CASE II 1995 PEAK HOUR	CASE III 1995 AVG. HOUR	MINIMAL CONFIGURATION CASE IV 1995 PEAK HOUR
1. WX RETRIEVAL	3.240	3.503	3.109	4.871
2. DISK I/O	2.025	2.029	2.021	2.029
3. OUTPUT	.021	.022	.020	.022
4. INTERPROCESSOR	.014	.014	.014	.014
TOTALS (SECONDS)	5.300	5.568	5.164	6.936

4.1.4.3.6 Local Briefing - Calculations and Final Results

1. First Page - Total response time for output of first page of local briefing is given in Table 4-11.

2. Remainder of Pages - Total time until rest of pages in local briefing are available for display is summarized in Table 4-12.

4.1.4.3.7 Variance of Response Time

In general, complex queueing models such as this will not predict response time variance sufficiently accurately to be relied upon.* However, Table 4-13 shows an estimate for output of the first page of the data, given the recommended configuration, 1995 demand, peak hour, assuming the central limit theorem can be invoked to give results that are at least approximately correct. Given a normal distribution, with mean $\mu = 2.664$ (from Table 4-9) and standard deviation $\sigma = .8$ (Table 4-13), approximately 85% of all route briefings will take less than $N + 1\sigma = 3.5$ seconds, better than 95% will take less than $M + 2\sigma = 4.3$ seconds, and 99% will take under $M + 3\sigma = 5.1$ seconds.

4.1.4.4 Response Times for Flight Plans, Aircraft Contacts, and Miscellaneous Messages

The response times for flight plans, aircraft contacts and miscellaneous messages is estimated in the following sections.

* W. Chang, "Single Server Queueing Process in Computing Systems," IBM Systems Journal, No. 1, 1970, p. 62.

TABLE 4-11

MEAN RESPONSE TIMES FOR FIRST PAGE OF LOCAL BRIEFING IN FUNCTION SHARING CONFIGURATION

ITEM	RECOMMENDED CONFIGURATION 1985 PEAK HOUR	RECOMMENDED CONFIGURATION 1995 PEAK HOUR	RECOMMENDED CONFIGURATION 1995 AVG. HOUR	MINIMAL CONFIGURATION 1995 PEAK HOUR
1. INPUT REQUEST	.040	.041	.039	.041
2. WX RETRIEVAL	.120	.120	.119	.122
3. DISK I/O	.132	.132	.131	.132
4. OUTPUT	.021	.022	.020	.022
5. INTERPROCESSOR	.028	.028	.028	.028
TOTALS (SECONDS)	.341	.343	.317	.345

TABLE 4-12

MEAN RESPONSE TIMES FOR RETRIEVAL OF REMAINDER OF DATA FOR LOCAL BRIEFING IN FUNCTION SHARING CONFIGURATION

ITEM	RECOMMENDED CONFIGURATION 1985 PEAK HOUR	RECOMMENDED CONFIGURATION 1995 PEAK HOUR	RECOMMENDED CONFIGURATION 1995 AVG. HOUR	MINIMAL CONFIGURATION 1995 PEAK HOUR
1. WX RETRIEVAL	.819	1.204	.654	4.280
2. DISK I/O	.658	.659	.656	.659
3. OUTPUT	.021	.022	.020	.022
4. INTERPROCESSOR	.014	.014	.014	.014
TOTALS	1.512	1.899	.344	4.975

TABLE 4-13

RESPONSE TIME VARIANCE

ITEM	X^2	$\bar{X}^2$	V(x)
1. INPUT($V(x) = \bar{x}^2$)	3.36×10^{-3}	1.68×10^{-3}	1.68×10^{-3}
2. ROUTE PROCESSING	5.618	5.11	.511
3. WX RETRIEVAL* ITEM (3)	4.92×10^{-4}	3.69×10^{-4}	1.23×10^{-4}
ITEM (4)	5.11×10^{-3}	3.83×10^{-3}	1.28×10^{-3}
ITEM (5)	2.31×10^{-3}	1.54×10^{-3}	7.7×10^{-4}
4. DISK I/O**	1.25×10^{-1}	1.74×10^{-2}	1.075×10^{-1}
5. OUTPUT ($V(X) = \bar{X}^2$)	9.68×10^{-4}	4.84×10^{-4}	4.84×10^{-4}
6. INTERPROCESSOR COMM.	1.76×10^{-3}	1.76×10^{-3}	0
			$.623 = \sigma^2$
			$.789 = \sigma \approx .8$

* DISTRIBUTION ASSUMED TO BE THAT IN TABLE 4-4

** ASSUMING $V(X) = (E(X))^2$ FOR SINGLE ACCESS AND USING FORMULA $V(CY) = C^2V(Y)$ WHERE C = 2.5

4.1.4.4.1 Miscellaneous Messages and Aircraft Contacts

Miscellaneous messages and Aircraft Contacts are assumed handled by the input processor alone, with the following breakdown of steps:

Aircraft Contact

Processing	20K instructions
Disk Accesses	2 overlapped

Miscellaneous Messages

Processing	10K instructions
Disk Accesses	5 overlapped

Before response time can be calculated, disk service time must be estimated. For 1995, peak hour, the number of accesses is:

Aircraft contacts:	189 @ 2 =	378
Logging	4500 @ 2 =	9000
Miscellaneous	3000 @ 5 =	15000
		24378 or about 24,000.

For an average hour, 1995, the number will be about 1/2 of this, or 12,000; and for 1985, peak hour, the number will be 2/3 of 24,000 or about 16,000. The same Tables as in Section 4.1.4.2.5 may be constructed, except here Cases II and IV are the same, since both have the same load and same number (one) of input processors. The calculations are summarized in Table 4-14.

TABLE 4-14

MEAN DISK ACCESS WAITING TIMES FOR DIFFERENT LOAD CONDITIONS AND DIFFERENT NUMBER OF DISKS/PROCESSOR

CASE I: RECOMMENDED CONFIGURATION, 1985 PEAK LOAD

NO. OF DISKS(m)	E(w)	Wq	B	ρ	AVG. ACCESSES/ DISK/HOUR
1	.0685	.0160	.2333	.2333	16,000
2	.0532	.0007	.0244	.1167	8,000
3	.0525	3.45×10^{-5}	.0018	.0778	5,333
4	.0525	1.45×10^{-6}	.0001	.0583	4,000

CASE II: RECOMMENDED CONFIGURATION, 1995 PEAK LOAD (ALSO, MINIMAL CONFIGURATION, 1995 PEAK LOAD)

NO. OF DISKS(m)	E(w)	Wq	B	ρ	AVG. ACCESSES/ DISK/HOUR
1	.0808	.0283	.3500	.3500	24,000
2	.0542	.0017	.0521	.1750	12,000
3	.0526	.0001	.0057	.1167	8,000
4	.0525	6.95×10^{-6}	.0005	.0875	6,000

CASE III: RECOMMENDED CONFIGURATION, 1995 AVERAGE LOAD

NO. OF DISKS(m)	E(w)	Wq	B	ρ	AVG. ACCESSES/ DISK/HOUR
1	.0636	.0111	.1750	.1750	12,000
2	.0529	.0004	.0141	.0875	6,000
3	.0525	1.48×10^{-5}	.0008	.0583	4,000
4	.0525	4.7×10^{-7}	3.43×10^{-5}	.0438	3,000

ρ = system utilization   Wq = queuing time

B = multiserver queue factor   E(w) = expected service time

Either three or four disks would be satisfactory for this load, the only significant difference being that under peak conditions somewhat greater redundancy would be available with four disks, since then two could fail with relatively little effect on response time. In addition, four would provide a greater margin of safety in respect of the utilization, the estimate for which could be low by 40-50% again with little effect on response time. So here four disks will be assumed for response time calculations.

Processing time is estimated as in the case of input request processing for local and route briefings, Section 4.1.4.3.1, except that the number of instructions to be executed is different. For example, the 1995 peak hour load calculation is

$$20K \times 1.6/420K + .006 = .0822$$

For the four cases under consideration, response time calculations are summarized in Tables 4-15 and 4-16.

4.1.4.4.2 Flight Plan Processing

Flight Plan Processing is very similar to Local Weather Retrieval Processing, first page, except that the flight plan processor need not be considered to have a preemptive resume queue discipline, but instead can be modelled simply as an M/G/1 system, and the Pollaczek-Khintchine equations employed. From the data

TABLE 4-15

MEAN RESPONSE TIMES FOR PROCESSING OF AIRCRAFT CONTACT MESSAGES IN FUNCTION SHARING CONFIGURATIONS

	RECOMMENDED CONFIGURATION			MINIMAL CONFIGURATION
ITEM	1985 PEAK HOUR	1995 PEAK HOUR	1995 AVG. HOUR	1995 PEAK HOUR
1. AC PROCESSING	.0783	.0793	.0773	.0793
2. DISK I/O	.1050	.1050	.1050	.1050
TOTAL (SECONDS)	.183	.184	.182	.184

TABLE 4-16

MEAN RESPONSE TIMES FOR PROCESSING OF MISCELLANEOUS MESSAGES IN FUNCTION SHARING CONFIGURATION

	RECOMMENDED CONFIGURATION			MINIMAL CONFIGURATION
ITEM	1985 PEAK HOUR	1995 PEAK HOUR	1995 AVG. HOUR	1995 PEAK HOUR
1. MISC MSG.	.0402	.0412	.0392	.0412
2. DISK I/O	.1313	.1313	.1313	.1313
TOTAL (SECONDS)	.171	.173	.170	.173

in Appendix A, the first and second moments of service time may be determined, assuming an E_{10} distribution, and are shown in Table 4-17.

Machine utilization for the four cases may be estimated using the data in Appendix A, $\rho = \lambda\bar{s}$; and with the foregoing data the expected queueing time determined. The results are summarized in Table 4-18.

Disk utilization may be estimated based on the data in Appendix A, and expected queueing and waiting times, in the same manner as before. It is assumed that flight plan processing requires 10 disk accesses. Again, Cases II and IV are the same for this task. The results are given in Table 4-19.

Clearly two disks on the Flight Plan processor will be adequate to handle the load, provide redundancy and assure a margin of safety. For the purpose of calculating response time, overlapped access will be assumed to reduce total disk access time by 30%.

Response time may now be evaluated for the four cases, and is summarized in Table 4-20.

TABLE 4-17

MOMENTS FOR FLIGHT PLAN PROCESSING

TYPE	1st MOMENT	2nd MOMENT
IFR	(.72)(100K)/22K = .2748	$(72)100K/262K)^2$ (11/10) = .1154
VFR	(28)(50K)/22K = .0534	$(28)(50K/262K)^2$ (11/10) = .0112
TOTAL	$\bar{s}$ = .3282	$\overline{s^2}$ = .1266

$$\sigma_s^2 = \overline{s^2} - (\bar{s})^2 = .1266 - (.3282)^2 = .0189$$

TABLE 4-18

E(q) FOR FLIGHT PLAN PROCESSING
IN FUNCTION SHARING CONFIGURATION

ITEM	RECOMMENDED CONFIGURATION 1985 PEAK HOUR	RECOMMENDED CONFIGURATION 1995 PEAK HOUR	RECOMMENDED CONFIGURATION 1995 AVG. HOUR	MINIMAL CONFIGURATION 1995 PEAK HOUR
ρ	.0267	.0378	.0189	.0378
E(q)	.005	.008	.004	.008

ρ = SYSTEM UTILIZATION　　　E(q) = EXPECTED QUEUEING TIME

TABLE 4-19

MEAN WAITING TIMES FOR DISK ACCESSES FOR FLIGHT PLAN PROCESSING IN FUNCTION SHARING CONFIGURATION

CASE I: RECOMMENDED CONFIGURATION, 1985 PEAK HOUR

NO. OF DISKS(M)	E(w)	Wq	B	ρ	AVG.ACCESSES/DISK/HR.
1	.0548	.023	.0427	.0427	2930
2	.0525	2.4×10^{-5}	.0009	.0214	1465

CASE II: RECOMMENDED CONFIGURATION, 1995 PEAK HOUR

NO. OF DISKS(M)	E(w)	Wq	B	ρ	AVG.ACCESSES/DISK/HR.
1	.0559	.0034	.0605	.0605	4150
2	.0525	4.8×10^{-5}	.0018	.0303	2075

CASE III: RECOMMENDED CONFIGURATION, 1995 AVERAGE HOUR

NO. OF DISKS(M)	E(w)	Wq	B	ρ	AVG.ACCESSES/DISK/HR.
1	.0541	.0016	.0303	.0303	2075
2	.0525	1.2×10^{-5}	.0005	.0151	1038

(SYMBOLS EXPLAINED ON PAGE 4-26)

TABLE 4-20

MEAN RESPONSE TIME FOR FLIGHT PLAN
PROCESSING IN FUNCTION SHARING CONFIGURATION

ITEM	RECOMMENDED CONFIGURATION 1985 PEAK HOUR	1995 PEAK HOUR	1995 AVG. HOUR	MINIMAL CONFIGURATION 1995 PEAK HOUR
1. INPUT REQUEST	.058	.060	.057	.060
2. FLIGHT PLAN PROC.	.333	.336	.332	.336
3. DISK I/O	.368	.368	.368	.368
4. OUTPUT	.021	.022	.020	.022
5. INTERPROCESSOR	.014	.014	.014	.014
TOTALS (SECONDS)	.794	.800	.791	.800

4.1.5 Cost Estimate (Per Hub - Recommended Configuration)

The cost estimate in Table 4-21 is for the data processing hardware and associated equipment. It does not include the cost for Specialist or other consoles.

The design analyzed and costed here is intentionally very conservative with respect to component utilization for the reason that large scale distributed systems are not yet in general use and this procedure should ensure adequate reserve capacity for unanticipated problems and higher than expected loading. The cost could be trimmed by using fewer processors or disks, for example, but the relatively small decrease in cost would not justify the increase in risk.

The 50% cost increment for expansion capacity is an estimate which, pending field experience, cannot be made more precise with respect to specific equipment that may need to be added.

TABLE 4-21

COST ESTIMATE PER HUB FOR RECOMMENDED
FUNCTION SHARING CONFIGURATION

T16/1412 PROCESSOR, 384K MEMORY 8 @ $64,500	=	$516,000
T16/6302 ASYNCHRONOUS EXTENSION BOARD 8 @ 4,300	=	34,400
T16/7501 TERMINAL CONNECTION PANEL 8 @ 775	=	6,200
T16/6301 ASYNCH CONTROLLER 8 @ 2,900	=	23,200
T16/6512 CRT 8 @ 2,400	=	19,200
T16/3102 DISC CONTROL 19 @ 4,800	=	91,200
T16/4102 MHD 50M BYTE 19 @ 14,500	=	275,500
T16/3201 MAG TAPE CONTROL 7 @ 3,800	=	26,600
T16/5101 MAG TAPE DRIVE 7 @ 7,400	=	51,800
T16/3301 LINE PRINTER CONTROLLER* 2 @ 1,800	=	3,600
T16/5502 LINE PRINTER 300 LPM* 2 @ 11,500	=	23,000
T16/7103 SYSTEM CABINET 3 @ 6,800	=	20,400
T16/2001 DECIMAL ARITHMETIC PACKAGE 8 @ 1,500	=	12,000
T16/3303 CARD READER CONTROLLER* 1 @ 1,800	=	1,800
T16/5301 CARD READER 600 CPM* 1 @ 4,800	=	4,800
Adding 50% for 50% expansion, total cost is $1,684,500.		$1,123,000

* Not included in system configuration diagram; may be attached to any machine.

4.2 Load Sharing Configuration

In a load sharing configuration there are several machines, each performing all functions of the Flight Service Station for a subset of the terminals. The same number of terminals is assumed connected to each processor. There may be off-loading of certain functions, such as communications and weather data base maintenance, to improve response time and reduce duplication of effort. This type of arrangement will be assumed here, for the reasons explained in Section 4.1.3. A discussion of the differences, advantages, and disadvantages of the load sharing and function sharing configurations will be found in Section 4.3. Here, as in Section 4.1, a preliminary design based on throughput considerations will be illustrated, followed by a design which is more optimal with reference to response time and sensitivity to higher than anticipated loads, either functional or numerical. Finally, a queueing analysis will be performed on both designs.

4.2.1 Assumptions

To analyze the performance of a split load distributed processing configuration, the following assumptions will be made:

1. Each machine performs all functions except data base maintenance and communications with outside world.
2. Data base maintenance will be done as described for the function sharing configuration in Section 4.1.3.

3. TANDEM computers will be used for sizing and costing purposes.

4. Speed of TANDEM machines assumed to be that of NAS instruction mix (420K IPS).

5. Overhead is assumed to be 60% of operational software.

6. FSS Hub CPU requirements are as in Appendix B, with modifications as noted.

7. No processing will begin until all required entries have been made.

The effect of relaxing assumption (7) will be discussed in Section 4.3.

The data flow for the three major FSS functions (Route briefing, Local briefing, and Flight plan filing) is the same as that described in Section 4.1.1.

4.2.2 Estimate of Number of Machines and Load/Machine

The number of instructions per second of raw computing power required may be determined using the data in Appendices A and B. This is done in Sections 4.2.2.1, 4.2.2.2, and 4.2.2.3. In Section 4.2.2.4 the effect of utilizing the special method of data base management described in Section 4.1.3 is considered.

4.2.2.1 Flight Services Throughput Calculation

At this point, an estimate of the required FSS Hub processing throughput, in instructions per second, can be made. The peak

hour number of instructions required for flight services can be calculated as follows:

Route Briefings

$$\# \text{ Inst./Sec.} = \left(\frac{606 \text{ route briefs}}{3600 \text{ sec}}\right)\left(1.121 \times 10^6 \ \frac{\text{instructions}}{\text{route brief}}\right)$$

$$= .188 \times 10^6$$

Local Briefings

$$\# \text{ Inst./Sec.} = \left(\frac{307 \text{ local briefs}}{3600 \text{ sec}}\right)\left(.092 \times 10^6 \ \frac{\text{instructions}}{\text{local brief}}\right)$$

$$= .008 \times 10^6$$

Flight Plans

$$\text{IFR Inst./Sec.} = \left(\frac{299 \text{ FPs}}{3600 \text{ sec}}\right)\left(0.100 \times 10^6 \ \frac{\text{instructions}}{\text{FP}}\right)$$

$$= .008 \times 10^6$$

$$\text{VFR Inst./Sec.} = \left(\frac{116 \text{ FPs}}{3600 \text{ sec}}\right)\left(0.050 \times 10^6 \ \frac{\text{instructions}}{\text{FP}}\right)$$

$$= .002 \times 10^6$$

Aircraft Contacts

$$\text{Inst./Sec.} = \left(\frac{189 \text{ AC messages}}{3600 \text{ sec}}\right)\left(0.020 \times 10^6 \ \frac{\text{instructions}}{\text{AC message}}\right)$$

$$= .001 \times 10^6$$

$$\boxed{\text{Flight Services Instructions Per Second} = .207 \times 10^6}$$

4.2.2.2 Additional Services Throughput Calculations

In addition to processing flight service demands, the Hub processor must receive about 1000 weather messages every four minutes

from the AWP during SA transmission, and also do miscellaneous message recording. This results in the following throughput requirements:

Weather Data

$$\text{Inst./Sec.} = \left(\frac{1000 \text{ messages}}{240 \text{ seconds}}\right)\left(.010 \times 10^{6} \ \frac{\text{instructions}}{\text{messages}}\right)$$

$$= .041 \times 10^{6}$$

Miscellaneous

$$\text{Inst./Sec.} = \left(\frac{16{,}517 \text{ messages}}{3600 \text{ seconds}}\right)\left(.010 \times 10^{6} \ \frac{\text{instructions}}{\text{messages}}\right)$$

$$= .046 \times 10^{6}$$

$$\boxed{\text{Additional Instructions Per Second} = .087 \times 10^{6}}$$

Where 16,517 = 606 + 307 + 299 + 116 + 189+ 15,000 messages per hour from flight service, weather data, and miscellaneous message estimates.

4.2.2.3 Total FSS Hub Throughput Calculation

Total FSS Hub Processor throughput is estimated as:

Number of Instructions Per Second =

$$(.207 \times 10^{6} + .087 \times 10^{6})\left(\frac{1}{0.6}\right)(1.6)\ (1.5) = 1.18 \times 10^{6}$$

The coefficients in the equation are estimates to account for the following:

1. $\left(\frac{1}{0.6}\right)$, 60% utilization of the CPUs processing capability;
2. (1.6), 60% operating system executive overhead; and
3. (1.5), 50% expansion capacity.

4.2.2.4 Effect of Using Recommended Method of DBM (See Paragraph 4.1.3)

In this case, the sum of instructions omits the $.041 \times 10^6$ instructions needed for weather data base update; and the number of miscellaneous messages per second is now 606 + 307 + 299 + 116 + 189 + 3000 + 4500 = 9017 and the total is thus:

$$.207 \times 10^6 + \frac{9017}{3600} \times .010 \times 10^6 = .232 \times 10^6$$

To estimate total computing power required, multiply by appropriate factors for CPU utilization, overhead, and expansion capacity:

$$232\text{K IPS} \times \frac{1}{6} \times 1.6 \times 1.5 = .93 \text{ MIPS}$$

CPU Utilization

Overhead

Expansion Capacity

(Expansion Capacity is included here, but not in Section 4.1, for reasons discussed in Section 4.1.2.2).

To determine number of processors needed (taking into account only throughput), first take number of instructions/sec. and divide by instruction speed, then round to next highest integer:

$$\frac{.93 \text{ MIPS}}{420\text{K IPS}} = 2.21, \text{ so 3 processors needed.}$$

The peak hour load per processor will then be $\frac{.93 \text{ MIPS}}{3} = 310\text{K IPS}$.

Next, the number of processors required for communication and data base maintenance must be estimated. Since the same technique is assumed as employed for the function sharing configuration, the number of processors will be one, operating at about 110K IPS (see Section 4.1.2 for details). Finally, one redundant processor will be required.

To determine average terminal load/processor, simply divide the number of terminals by the number of processors servicing terminals:

$$\frac{\text{80 terminals}}{\text{3 processors}} = \text{27 terminals/processor}$$

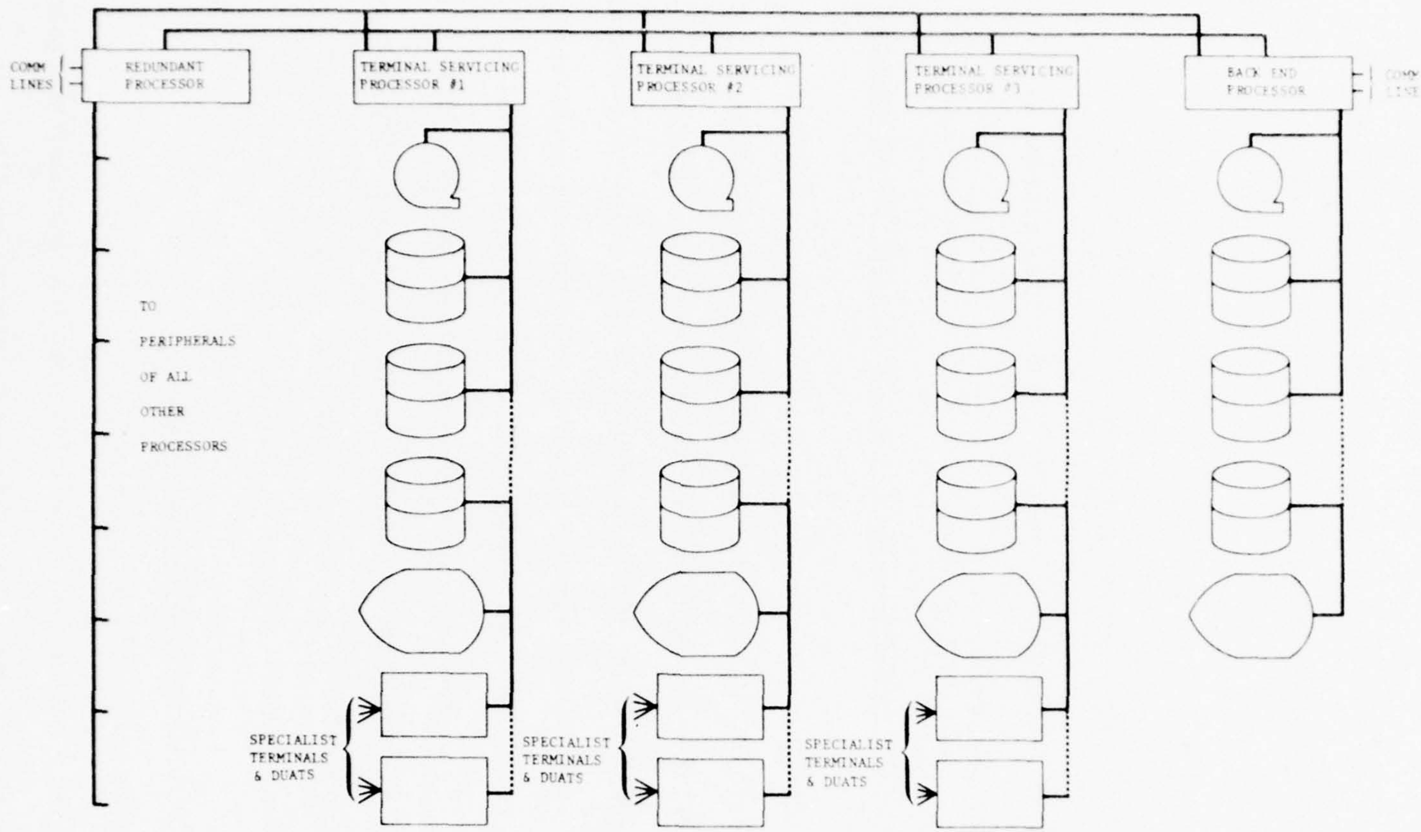
1M byte of main storage is assumed for each processor servicing terminals, 1M byte on the redundant processor, and 384K on the back end processor.

Peripheral assignment is as follows:

Each processor will have 3 moving head disks with one spare for redundancy. Each machine will also have 1 tape drive, but because the tape drive is not used for on-line processing, it is not duplicated.

4.2.3 Preliminary Configuration and Recommended Configuration

A tentative configuration based on the foregoing considerations is shown in Figure 4-8. Machine utilization (excluding the CPU utilization factor and the expansion factor) is indicated.



	1M BYTE	1M BYTE	1M BYTE	1M BYTE	384K BYTE
AVERAGE INSTRUCTIONS EXECUTED/ SECOND		310K	310K	310K	110K
MACHINE UTILIZATION		.28	.28	.28	.16

FIGURE 4-8
PRELIMINARY LOAD SHARING CONFIGURATION
USING TANDEM PROCESSORS

Improvement in system performance and fail-soft capabilities may be realized by addition of one more terminal handling processor. Such a configuration is shown in Figure 4-9. The peak hour load per processor is:

$$\frac{.93 \text{ MIPS}}{4} = 233\text{K IPS}$$

The number of terminals per processor will be $\frac{80}{4} = 20$.

The expected performance of both of these configurations is analyzed in Section 4.2.4.

4.2.4 Queueing Considerations For a Load Sharing FSS Configuration

4.2.4.1 General

In a load sharing distribution processing configuration, each machine performs all functions of the FSS except processing of incoming weather data, which is assumed to be handled by a separate machine. Each of the terminal handling processors is physically connected to a number of terminals, and manages all of their requests. Since the tasks vary considerably in importance and length, as discussed in Section 4.1.4, each processor will utilize an M/G/1 Preemptive-resume queueing scheme. Here 12 levels of priority are assumed, as shown in Figure 4-10.

4.2.4.2 Analysis of System Under Load

The following cases will be analyzed:

1. Case I - Peak hour, 1985, four terminal handling processors.
2. Case II - Peak hour, 1995, four terminal handling processors.

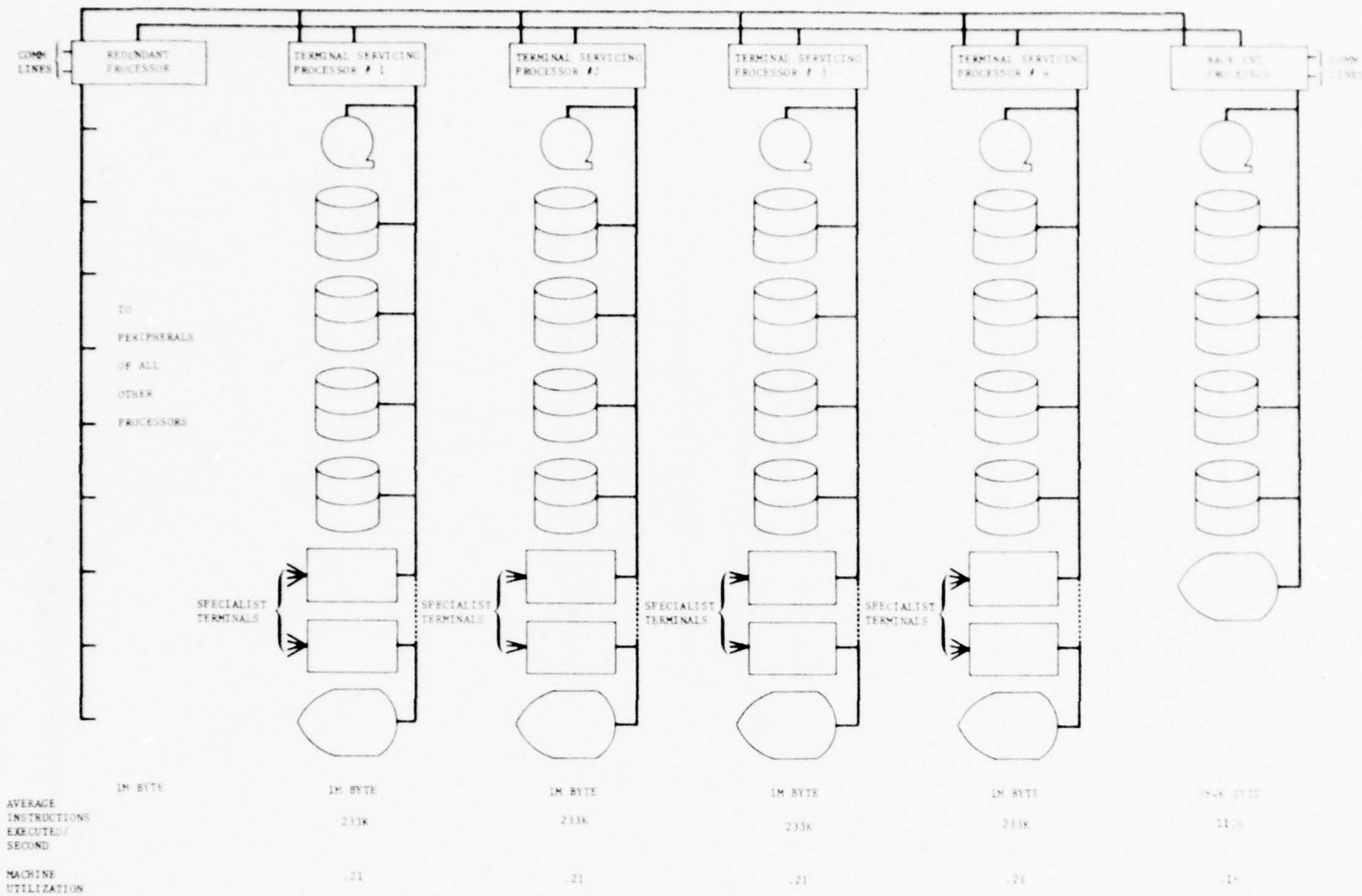


FIGURE 4-9
RECOMMENDED LOAD SHARING CONFIGURATION EMPLOYING TANDEM PROCESSORS

CPU

- (1) Processor status messages
- (2) Data base update
- (3) Input from terminals - all categories
- (4) Output to terminals - weather data, pages
- (5) Output to terminals - other categories
- (6) I/O ready list, disk processing, weather retrieval - first page
- (7) Route processing
- (8) Flight Plan Processing
- (9) Aircraft contact processing
- (10) Logging and Miscellaneous messages
- (11) I/O ready list, rest of route brief
- (12) I/O ready list, rest of local brief

FIGURE 4-10
CATEGORIES IN LOAD SHARING CONFIGURATION

3. Case III - Average hour, 1995, four terminal handling processors.

4. Case IV - Peak hour, 1995, three terminal handling processors.

For each of these cases, the load/processor/hour is given in Table 4-22, based on Appendix A.

4.2.4.2.1 CPU Analysis

In order to determine the queueing and waiting times for the 12 categories in each of the four cases, the first and second moments of service time distribution must be known. These are tabulated in Table 4-23 with the 12 individual categories given in the left column, together with their respective first and second moments ($\alpha^{(1)}$ and $\alpha^{(2)}$), and the corresponding value of λ (poisson arrival rate) for each of the four cases.

Waiting times are determined by using the formulae for a preemptive-resume queueing model. The results of these calculations are summarized in Table 4-24 for the 12 categories in each of the four cases. Also given is the service time, to facilitiate direct comparison with the estimated mean waiting time. All numbers are given to three significant figures.

4.2.4.2.2 Disk Analysis

Disk queueing must now be analyzed. The same assumptions as in Section 4.1.4.2.5 are made with respect to the disks, and again

TABLE 4-22

PROCESSOR LOAD FOR DIFFERENT DEMAND CONDITIONS

	ROUTE BRIEFING	LOCAL BRIEFING	FLIGHT PLAN	AIRCRAFT CONTACT
Case I	104	51	73	36
Case II	152	77	104	42
Case III	76	38	52	23
Case IV	202	102	138	63

TABLE 4-23

MOMENTS AND INTERARRIVAL RATES (λ) FOR DATA PROCESSING CATEGORIES

	λ for Case I	II	III	IV
1. Processor status messages @ 75 instr. each Service time assumed constant $\alpha 1^{(1)}$ = .0003 sec $\alpha_1^{(2)} = 8 \times 10^{-8}$ sec^2	6/sec	6/sec	6/sec	5/sec
2. Data base update* @ 500 instr. each E_3 service time distr. $\alpha 2^{(1)}$ = .0019 sec $\alpha 2^{(2)} = 4.9 \times 10^{-6}$ sec^2	90/hr = .025/sec	90/hr = .025/sec	90/hr = .025/sec	90/hr = .025/sec
3. Input from terminals all categories @ 10K instr. E_3 service time distr. $\alpha 3^{(1)}$ = .0382 sec. $\alpha 3^{(2)}$ = .0019 sec^2	753/hr = .2092/sec	1129/hr = .3137/sec	564/hr = .1568/sec	1506/hr= .4182/sec
4. Output to terminals Weather data, pages** Flight plans, pages*** @ 5K instr. E_{10} service time distr. $\alpha 4^{(1)}$ = .019 $\alpha 4^{(2)}$ = .004 sec^2	1656/hr = .4600/sec	2425/hr = .6737/sec	1214/hr = .3372/sec	3222/hr = .8950/sec

*See note on this item in Section 4.1.4.2.4.

**Assumes weighted average of 13.75 pages per route brief, 3 pages per local brief.

***Assumes one page per flight plan.

TABLE 4-23
(CONTD)

λ for Case

	I	II	III	IV
5. Output to terminals, other Categories @ 1K instructions E_2 service time distri. $\alpha_5^{(1)} = .0038$ $\alpha_5^{(1)} = 2 \times 10^{-5}$	506/hr = .1405/sec	758/hr = .2107/sec	379/hr - .1053/sec	1011/hr = .2809/sec
6. I/O ready list & processing First page (route & local briefs) @ 16K instructions E_3 service time distr. $\alpha_6^{(1)} = .0611$ sec $\alpha_6^{(2)} = .0050$ sec^2	154/hr = .0428/sec	228/hr = .0633/sec	114/hr = .0317/sec	304/hr = .0844/sec
7. Flight Plan Processing @ 86K instructions* $\alpha_7^{(1)} = .328$ $\alpha_7^{(2)} = .3558$	73/hr = .0203/sec	104/hr = .0289/sec	52/hr = .0144/sec	138/hr = .0383/sec
8. Aircraft Contact Processing @ 20K instructions E_3 service time distr. $\alpha_8^{(1)} = .0763$ sec $\alpha_8^{(2)} = .0078$ sec^2	36/hr = .0100/sec	47/hr = .0131/sec	23/hr = .0064/sec	63/hr = .0175/sec

*See Section 4.1.4.4.2 of previous analysis for moments and distributions.

TABLE 4-23
(CONTD)

λ for Case

	I	II	III	IV
9. Logging and Miscellaneous messages @ 9500 instructions E_2 service time distr. $\alpha_9^{(1)} = .0363$ sec $\alpha_9^{(2)} = .0020\ sec^2$	1250/hr = .3472/sec	1875/hr = .5208/sec	938/hr = .2606/sec	2500/hr = .6944/sec
10. Route Processing @ 434K instructions E_{10} service time distr. $\alpha_{10}^{(1)} = 1.66$ sec $\alpha_{10}^{(2)} = 3.02\ sec^2$	104/hr = .0289/sec	152/hr = .0422/sec	76/hr = .0211/sec	202/hr = .0561/sec
11. I/O ready list - rest of route brief @ 671K instructions E_2 service time distr. $\alpha_{11}^{(1)} = 2.56$ sec $\alpha_{11}^{(2)} = 9.84\ sec^2$	104/hr = .0289/sec	152/hr = .0422/sec	76/hr = .0211/sec	202/hr = .0561/sec
12. I/O ready list - rest of local brief @ 76K instructions E_3 service time distr. $\alpha_{12}^{(1)} = .290$ sec $\alpha_{12}^{(2)} = .112\ sec^2$	51/hr = .0142/sec	77/hr = .0214/sec	38/hr = .0106/sec	102/hr = .0283/sec

TABLE 4-24

MEAN CPU WAITING TIMES FOR LOAD SHARING CONFIGURATION

CATEGORY	SERVICE TIME	CASE I	CASE II	CASE III	CASE IV
1. Status Messages	.000300	.000300	.000300	.000300	.000300
2. Data Base Update	.00190	.00190	.00190	.00190	.00190
3. Terminal Input	.0382	.0385	.0386	.0384	.0387
4. Output, Weather Data, FP's	.0190	.0203	.0210	.0200	.0216
5. Output, Other Categories	.00380	.00504	.00564	.00470	.00629
6. First Page Processing	.0611	.0636	.0647	.0629	.0660
7. Flight Plan Processing	.328	.340	.346	.3371	.352
8. Aircraft Contact Processing	.0763	.0837	.0872	.0816	.0910
9. Miscellaneous Message Processing	.0363	.0430	.0462	.0411	.0499
10. Route Processing	1.66	1.79	1.85	1.75	1.93
11. Rest of Route Brief Processing	2.56	3.06	3.37	2.91	3.75
12. Rest of Local Brief Processing	.290	.622	.868	.511	1.23

a multi-server queueing model is employed. The number of disks accesses/hour for the four cases is itemized by category in Table 4-25.

For each of the four cases, the expected queueing and waiting times are determined, assuming one to four disks per machine (see Table 4-26).

In all cases four disks per processor seems to be the best choice both on account of the redundancy it provides and the margin of safety with respect to utilization.

4.2.4.2.3 Response Time Calculation

Using the foregoing data, response time for seven categories of interest may be determined. In all cases except the route processor it is assumed that overlapped disk accesses will reduce disk access time by a factor of two. The results are summarized in Table 4-27.

4.2.4.3 Comparison of The Response Times of Function Sharing And Load Sharing Configurations

In general, the performance of the two recommended configurations is very similar and is summarized for the 1995 peak hour load in Table 4-28. Route brief processing time, for example, is within 1% for first page output. The only items showing any relatively large deviation are those that involve interprocessor transfers on the function sharing configuration, and are of short run times.

TABLE 4-25

DISK ACCESS PER HOUR PER PROCESSOR FOR LOAD SHARING CONFIGURATION

ITEM	CASE I	CASE II	CASE III	CASE IV
1. Data Base Update	360 Per Disk	360 Per Disk	360 Per Disk	360 Per Disk
2. Route Brief Weather Retrieval @ 82	8528	12464	6232	16564
3. Local Weather Retrieval @ 30	1530	2310	1140	3060
4. Route Processing @ 11	1144	1672	836	2222
5. Flight Plan Processing @ 10	730	1040	520	1380
6. Aircraft Contact @ 2	72	94	46	126
7. Miscellaneous @ 5	2500	3750	1875	5000
TOTAL	14504 + 360 Per Disk	21330 + 360 Per Disk	10649 + 360 Per Disk	28352 + 360 Per Disk

TABLE 4-26

MEAN DISK ACCESS WAITING TIMES FOR DIFFERENT LOADS AND DEMANDS

CASE I: 1985 PEAK HOUR, FOUR TERMINAL HANDLING PROCESSORS

NUMBER OF DISKS	E(W)	W(q)	B	ρ	AVERAGE ACCESS/ DISK/HR
1	.0670	.0145	.2168	.2168	14864
2	.0532	.0007	.0222	.1110	7612
3	.0525	3.19×10^{-5}	.0017	.0758	5195
4	.0525	1.43×10^{-6}	.0001	.0581	3986

CASE II: PEAK HOUR - 1995 - FOUR TERMINAL HANDLING PROCESSORS

NUMBER OF DISKS	E(W)	W(q)	B	ρ	AVERAGE ACCESS/ DISK/HR
1	.0768	.0243	.3163	.3163	21690
2	.0539	.0014	.0445	.1608	11025
3	.0526	9.25×10^{-5}	.0047	.1089	7470
4	.0525	5.68×10^{-6}	.0004	.0830	5693

E(W) = MEAN SERVICE TIME

W(q) = MEAN QUEUEING TIME

B = PROBABILITY ALL SERVERS BUSY

ρ = SYSTEM UTILIZATION

TABLE 4-26
(CONTD)

CASE III: AVERAGE HOUR - 1995 - FOUR TERMINAL HANDLING PROCESSORS

NUMBERS OF DISKS	E(W)	Wq	B	ρ	AVERAGE ACCESS/ DISK/HR
1	.0625	.0100	.1605	.1605	11009
2	.0529	.0004	.0127	.0829	5685
3	.0525	1.38×10^{-5}	.0007	.0570	3910
4	.0525	4.85×10^{-7}	3.53×10^{-5}	.0441	3022

CASE IV: PEAK HOUR - 1995 - THREE TERMINAL HANDLING PROCESSORS

NUMBER OF DISKS	E(W)	Wq	B	ρ	AVERAGE ACCESS/ DISK/HR
1	.0903	.0378	.4187	.4187	28712
2	.0550	.0025	.0742	.2120	14536
3	.0527	.0002	.0100	.1431	9811
4	.0525	1.59×10^{-5}	.0011	.1086	7448

TABLE 4-27

MEAN RESPONSE TIMES (SECONDS) FOR TASKS IN LOAD SHARING CONFIGURATION

CATEGORY	CASE I 1985 PEAK HOUR 4 PROCESSORS	CASE II 1995 PEAK HOUR 4 PROCESSORS	CASE III 1995 AVG. HOUR 4 PROCESSORS	CASE IV 1995 PEAK HOUR 3 PROCESSORS
1. Route Briefing, Processing to				
output first page: (Items 3, 4, 6, 10)	1.912	1.974	1.871	2.056
11 serial, 5 overlapped disk accesses	.709	.709	.709	.710
	2.621	2.683	2.580	2.766
2. Local briefing, processing, to				
output first page (Items 3, 4, 6)	.122	.124	.121	.126
5 overlapped disk accesses	.131	.131	.131	.132
	.253	.255	.252	.258
3. Flight Plans				
(Items 5, 7) - Processing	.345	.352	.342	.358
10 overlapped disk accesses	.263	.263	.263	.263
	.608	.615	.605	.621
4. Aircraft Contacts				
(Items 5, 8) - Processing	.089	.093	.086	.097
2 disk accesses	.105	.105	.105	.105
	.194	.198	.191	.202
5. Miscellaneous messages				
(Items 5, 9) - Processing	.048	.052	.046	.056
5 overlapped disk accesses	.131	.131	.131	.132
	.179	.183	.177	.188

TABLE 4-27
(CONTD)

CATEGORY	CASE I 1985 PEAK HOUR 4 PROCESSORS	CASE II 1995 PEAK HOUR 4 PROCESSORS	CASE III 1995 AVG. HOUR 4 PROCESSORS	CASE IV 1995 PEAK HOUR 3 PROCESSORS
6. Remainder of Route Briefing				
(Item 11) - Processing	3.06	3.37	2.91	3.75
77 overlapped disk accesses	2.02	2.02	2.02	2.03
	5.08	5.39	4.93	5.78
7. Remainder of Local Briefing				
(Item 12) - Processing	.622	.868	.511	1.23
25 overlapped disk accesses	.656	.656	.656	.658
	1.278	1.524	1.167	1.888

TABLE 4-28

PERFORMANCE SUMMARY FOR RECOMMENDED FUNCTION SHARING AND LOAD SHARING CONFIGURATIONS, 1995 PEAK HOUR LOAD

CATEGORY	FUNCTION SHARING CONFIGURATION	LOAD SHARING CONFIGURATION	LARGE MAIN FRAME
1. Route Briefing, First Page	2.664*	2.683* (sec)	1.02
2. Local Briefing, First Page	.343	.255	.155
3. Flight Plan Processing	.800	.615	.330
4. Aircraft Contact Processing	.184	.198	.126
5. Miscellaneous Messages	.173	.183	.145
6. Remainder of Route Briefing	5.568	5.39	2.42
7. Remainder of Local Briefing	1.899	1.524	.785

*Mean drops by 1.2-1.5 seconds if route processing begins as soon as route is entered; see Section 4.3.1.

Here overhead causes the function sharing configuration to be 100-200 msec. slower (e.g., local briefing, flight plan filing). These differences, however, cannot be regarded as significant due to the short response times involved (less than one second).

For both configurations, the fact that the Case IV numbers (minimal hardware) shows a relatively small degree of response time degration for nearly all functions indicates that the systems have a comfortable margin of safety designed in with respect to performance.

For comparison purposes, response time estimates for two large main frame configurations are presented in Appendix C, and for one of the two is included in Table 4-28.

4.2.5 Cost Estimate (Per Hub, Recommended Configuration of 4 Terminal Handling Processors)

The cost estimate in Table 4-29 is for the data processing hardware and associated equipment, and does not include the cost of consoles for use by the Specialists or supervisory personnel.

TABLE 4-29

COST ESTIMATE PER HUB FOR RECOMMENDED LOAD SHARING CONFIGURATION

T16/1416	Processor 512K Bytes	5 @	$82,500 =	$ 412,500
	Semic Storage 512K Bytes	5 @	72,000 =	360,000
T16/7501	Terminal Connection Panel	10 @	775 =	7,750
T16/6301	Asynchronous Controller	10 @	4,300 =	43,000
T16/6302	Asyncrhonous Extension	10 @	4,300 =	43,000
T16/6512	CRT	6 @	2,400 =	14,400
T16/3102	Disk Control	20 @	4,800 =	96,000
T16/4102	MHD 50M Byte	20 @	14,500 =	290,000
T16/3201	Mag. Tape Control	5 @	3,800 =	19,000
T16/5101	Mag. Tape Drive	5 @	7,400 =	37,000
T16/1412	Processor 384K Bytes	1 @	64,500 =	64,500
	Switch/Patch Panel	1 @	50,000 =	50,000
T16/3301	Line Printer Controller*	2 @	1,800 =	3,600
T16/5502	Line Printer 300 LPM*	2 @	11,500 =	23,000
T16/7103	System Cabinet	2 @	6.800 =	13,600
T16/2001	Decimal Arithmetic Package	6 @	1,500 =	9,000
T16/3303	Card Reader Controller *	1 @	1,800 =	1,800
T16/5301	Card Reader 600 CPM*	1 @	4,800 =	4,800
				$1,492,950

*Not included in system configuration diagram; may be attached to any machine.

4.3 Comparison and Evaluation of Load Sharing and Function Sharing Designs

As the queueing analyses of Sections 4.1.4 and 4.2.4 demonstrate, there is relatively little difference between the function sharing and load sharing distributed processing configurations with respect to performance. Moreover, both are fail-safe, so that a loss of one element will have no effect. The primary differences between the configurations will be the difference in development required for each, and the difference in their fail-soft performance.

The load-sharing configuration is the closest to what already exists in operational form, and involves the least risk from the standpoint of being the least "distributed" of the two configurations. Its capacity (with respect to number of terminals handled per Hub) can easily be expanded by addition of complete subsystems of terminal handling processors and peripherals. It is also easily adapted to Hubs of different loads by changes in the number of terminal handling processors. If the functional capacity of the machine should ever become overtaxed, additional machines could be added and the load (number of terminals) on each reduced. In fail-soft mode, this type of system will degrade by loss of some number of terminals.

The function sharing configuration is the more elegant of the two, and more in the general direction that distributed processing

systems appear to be heading. In addition, its software would of necessity have to be modular, and this would facilitate structured programming. Additional functions (e.g., voice response systems) could easily be accommodated by addition of a new machine dedicated to the task. Additional machines could easily be added as route processors or terminal interface machines, for example, if these prove to be system bottlenecks. Allowing additional machines for an expansion capacity equivalent to that of the load sharing configuration, the function sharing configuration turns out to be somewhat more expensive hardware-wise per Hub.

4.3.1 Effect on Response Time of Relaxing Assumption (7)

Assumption (7), Sections 4.1.1 and 4.2.1, forbade the data processing system from starting to work on a job until all entries have been completed. In some cases, notably for route briefings, it may be of some advantage to begin processing before all entries are completed. In particular, as soon as a route, e.g., LAX-IAD, is detected, the route processor could be initiated. Although it is difficult to estimate precisely how much impact this would have on overall response times, the following breakdown appears reasonable: (1) In 25% of cases, the Specialist either enters the route last, or enters no other information and simply requests weather between two points; (2) In 75% of cases, one or more additional lines of information will be typed in. Since this will generally take at least two seconds, in nearly all cases the route processor will have finished its task, and so its time

will drop out of the overall response time figure. Hence, mean response times for route briefings (first page) will drop by about 1.5 seconds, to about 1.2-1.3 seconds. But of course this mean figure is not representative of actual response time, which will either remain at about 2.7 seconds or drop to about .4 seconds.

There appears to be no engineering reason why processing could not be done in this manner, as in fact it currently is in the AWANS system.

5. BRIEFINGS BY VENDORS

During the course of the MITRE/FAA Distributed Processing study of March 1977 led by ARD-440, several vendors gave briefings on their products, with specific reference to distributed processing:

1. Raytheon Corporation
2. Digital Equipment Corporation
3. Data General Corporation
4. Interdata Corporation
5. Tandem Corporation
6. Texas Instruments Corporation
7. Burroughs Corporation

Several things became apparent during the course of these briefings: (a) only one manufacturer has addressed the problem of fail-safe head-on, i.e., designed and built a system from the ground up, integrating both hardware and software so as to cover all potential single points of failure, provide special dual power supplies, etc., (b) other manufacturers have assembled the hardware for multiprocessing systems, but non offers fail-safe capabilities or features off-the-shelf, nor did any point to distributed processing installations we might visit. Only the one company offered to release the names of all its customers and invited the study team to visit whomever it chose to verify its claims, (c) in general, the minicomputer manufacturers are still geared to dealing with the customers who buy minicomputers

because they are cheap, and who wish to squeeze all available computing power out of them; this of course is incompatible with good response time in systems with random inputs (and the design goals of the FSS program generally), but apparently that consideration is of little or no concern to most of these customers. Even some of the distributed processing configurations appeared to be designed with the same goal of maximizing throughput in mind. Clearly for most minicomputer manufacturers, the hardware associated with fail-safe operation would, for the average customer, be an uneconomical and uncompetitive extravagence.

APPENDIX A

DEMAND ESTIMATES FOR 1985, 1995*

A.1 1985 - Cleveland Hub (Worst Case; Peak Hour)

A.1.1 Peak Hour Statistics

1. Pilot Briefs - 616
2. Flight Plans - 292
3. Aircraft Contacts - 145

A.1.2 Calculations of IFR and VFR Flight Plans

72% of flight plans are IFR = (293)(.72) = 211

VFR = Total - IFR = 293 - 211 = 82

A.1.3 Calculation of Route Oriented and Local Briefings

All IFR briefings are route oriented	= 211
50% of VFR briefings are route oriented = (.5)(616-211)	= 203
Total route oriented briefs	414

Total local briefs = Total briefs - route briefs = 616-414 = 202

A.2 1995 - Cleveland Hub (Worst Case; Peak Hour)

A.2.1 Peak Hour Statistics

a. Pilot Briefs - 913

b. Flight Plans - 415

c. Aircraft Contacts - 189

* Based on Reference 5, Appendix C.

A.2.2 Calculation of IFR and VFR Flight Plans

72% of flight plans are IFR = (415)(.72) = 299

VFR = Total - IFR = 415 - 299 = 116

A.2.3 Calculations of Route Oriented and Local Briefings

All IFR briefings are route oriented = 299

50% of VFR briefings are route oriented = (.5)(913-299) = 307

Total route oriented briefs 606

Total local briefs = Total briefs - route briefs =

913 - 606 = 307

A.3 Average Hour Demand

Based on the tables and graphs in MITRE MTR-6607, Vol. II, FSS Configuration Analysis Study, average hour demand is very close to half of the peak hour demand.

A.4 Demand Summary

Below in Table A-1 is a summary of peak and average hour demand for 1985 and 1995.

A.5 Specialist/DUAT Split

For 1995, the Route Briefings are assumed to be split as follows:

25% Specialist

75% Pilot at DUAT

This yields

(.25)(606) = 152 Specialist Briefings

(.75)(606) = 459 Pilot self-briefings

TABLE A-1

DEMAND SUMMARY FOR 1985, 1995

	1985		1995	
	AVERAGE	PEAK	AVERAGE	PEAK
IFR FLIGHT PLANS	106	211	150	299
VFR FLIGHT PLANS	41	82	58	116
ROUTE ORIENTED BRIEFS	207	414	303	606
LOCAL BRIEFS	101	202	154	307
AIRCRAFT CONTACTS	73	145	95	189
MISCELLANEOUS MESSAGES*	1000	2000	1500	3000
LOGGING MESSAGES	1528	3054	2260	4517

* ASSUMED TO BE APPROXIMATELY DOUBLE THE SUM OF THE OTHER MESSAGES.

A.6 Miscellaneous Messages, Logging Messages

Miscellaneous messages and logging messages are assumed to be as shown in Table A-1.

APPENDIX B

ESTIMATE OF INSTRUCTIONS EXECUTED AND DISK ACCESSES REQUIRED FOR FSS SYSTEM FUNCTIONS

B.1 Summary of Data Used in Report

Table B-1 summarizes the instructions required for a typical low altitude route oriented briefing. Table B-2 does the same for a typical local briefing. The data in these tables are based on information presented in Section B.2. Table B-3 summarizes the instructions required for other classes of FSS activity.

B.2 Origin of Data Used in Estimates of Instructions for FSS Functions in Section B.1

In this section, the data on which the figures in Section B.1 are based are given, including instructions for file retrieval by weather type (Table B-4), average number of items retrieved by type (Table B-5), winds and temperatures aloft calculation data (Table B-6), data base characteristics and disk access counts by weather type (Section 2.1), and instructions executed during route processing (Section 2.2). For some operations, however, hard data are not available, and estimates had to made (e.g., line and area filtering).

B.2.1 MITRE Experimental PSBT System - Converted Data Base Characteristics

B.2.1.1 SA

A. File Organization - Sets of three logical records arranged alphabetically to correspond to PSBCMI array SALOC. Each set of

TABLE B-1

LOW ALTITUDE ROUTE ORIENTED WEATHER BRIEFING INSTRUCTIONS ($x10^6$)

ITEM	PROCESSING FUNCTION	(# OF UNITS)	x	INSTRUCTIONS PER UNIT	=	(SUBTOTAL)	+	OVERHEAD INSTRUCTIONS	=	TOTAL INSTRUCTIONS	DISK ACCESSES
(a)	WEATHER LOCID RETRIEVAL FOR AN AVERAGE ROUTE LENGTH	480 nautical miles		.00018		.0864		.34800		.4344	11
(b)	GRAPHIC CHART RETRIEVAL	4 CHARTS		.00500		.0200		.00000		.0200	5
(c)	WARNING DATA RETRIEVAL (WSs, WAs, WHs, WACs, AND UWS)	2.6 WARNINGS		.00600		.0156		.00450		.0201	4
(d)	DENSITY ALTITUDE RETRIEVAL	1 RETRIEVAL		.01000		.0100		.00000		.0100	1
(e)	HOURLY WEATHER OBSERVATION (SA) RETRIEVAL	11.6 SAs		.00375		.0435		.00000		.0435	12
(f)	PILOT REPORTS RETRIEVED BY LOCID	.9 UA		.00675		.0061		.00900		.0151	2
	PILOT REPORTS CHECKED BY LINE FILTERING	15 UAs		.01000		.1500		.00000		.1500	1
(g)	TERMINAL FORECASTS WITH ETA FILTERING	5.4 FTs		.00350		.0189		.00000		.0189	6
(h)	WINDS AND TEMPERATURES ALOFT AND FLIGHT LOG	5 PONTS		.00500		.0250		.02600		.0510	6
(i)	RADAR REPORTS	2 SDs		.00350		.0070		.00000		.0070	3
(j)	NOTAMS RETRIEVED BY LOCID	6.2 NOTAMS		.00425		.0264		.00300		.0294	20

TABLE B-1
(CONTD)

ITEM	PROCESSING FUNCTION	(# OF UNITS) x	INSTRUCTIONS PER UNIT =	(SUBTOTAL) +	OVERHEAD INSTRUCTIONS =	TOTAL INSTRUCTIONS	DISK ACCESSES
(k)	NFDC/CARF NOTAM RETRIEVED BY LOCID	1 NOTAM	.00425	.0043	.00300	.0073	14
	RETRIEVED BY AREA FILTERING	2 AREAS	.05000	.1000	.00000	.1000	2
(l)	CENTRAL FLOW CONTROL DATA RETRIEVED BY LOCID	1 RETRIEVAL	.00425	.0043	.00300	.0073	1
	RETRIEVED BY AREA FILTERING	2 AREAS	.05000	.1000	.00000	.1000	2
(m)	MILITARY OPERATIONS DATA RETRIEVED BY LOCID	1 RETRIEVAL	.00425	.0043	.00300	.0073	1
	RETRIEVED BY AREA FILTERING	2 AREAS	.05000	.1000	.00000	.1000	2

TOTAL INSTRUCTIONS = 1.1213×10^6

TOTAL DISK ACCESSES = 93

TOTAL DISK ACCESSES EXCLUDING WEATHER LOCID RETRIEVAL (ROUTE PROCESSING) = 82

TABLE B-2

LOCAL WEATHER BRIEFING INSTRUCTIONS

INSTRUCTIONS IN MILLIONS ($x10^6$)

ITEM	PROCESSING FUNCTION	(# OF UNITS) x	INSTRUCTIONS PER UNIT	= (SUBTOTAL) +	OVERHEAD INSTRUCTIONS =	TOTAL INSTRUCTIONS	DISK ACCESSES
(a)	DENSITY ALTITUDE	1 RETRIEVAL	.01000	.0100	.00000	.0100	1
(b)	NOTAMS RETRIEVED	2 NOTAMS	.00425	.0085	.00300	.0115	15
(c)	WARNING DATA RETRIEVAL (WSs, UWs, WAs, WACs, AND WHs)	2 WARNINGS	.00600	.0120	.00450	.0165	3
(d)	HOURLY WEATHER OBSERVATIONS	3 SAs	.00375	.0113	.00000	.0113	3
(e)	PILOT REPORTS	3 UAs	.00675	.0203	.00900	.0293	[illegible]
(f)	TERMINAL FORECASTS	3 FTs	.00350	.0105	.00000	.0105	3
(g)	TWEB SYNOPSES	1 SYNOPSES	.00300	.0030	.00000	.0030	1

TOTAL INSTRUCTIONS = $.0921 \times 10^6$

TOTAL DISK ACCESSES = 30

TABLE B-3

INSTRUCTIONS REQUIRED FOR OTHER CLASSES OF FSS ACTIVITY

FLIGHT PLANS

IFR FLIGHT PLANS (SOME FORMAT CHECKING)	$= 0.10 \times 10^6$
VFR FLIGHT PLANS	$= 0.05 \times 10^6$

AIRCRAFT CONTACTS

AIRCRAFT CONTACT MESSAGE	$= 0.02 \times 10^6$

WEATHER DATA

UPDATE ONE WEATHER REPORT	$= 0.01 \times 10^6$

MISCELLANEOUS

RECORDING ONE MESSAGE AMENDMENT MESSAGES DEPARTURE MESSAGES PROGRESS REPORT MESSAGES FLIGHT PLAN READOUTS LIST DISPLAYS AFOS GRAPHICS UPDATE WEATHER RADAR DIAL UP AND DISPLAY UPDATE SA/NO ENTRY PILOT REPORT ENTRY TWEB/PATWAS PREPARATION INTERFACILITY TRANSMISSION AV-AWOS PROCESSING SUPERVISORY INPUTS	$= 0.01 \times 10^6$

TABLE B-4

ESTIMATED INSTRUCTIONS FOR WEATHER FILE RETRIEVAL

MESSAGE TYPE	ESTIMATED OVERHEAD INSTRUCTIONS			ESTIMATED ACCESS INSTRUCTIONS			TOTAL AVERAGE
	PDP 11/70 RSX-11D	370/145 CMS/OS	AVERAGE	PDP 11/70 RSX-11D	370/145 CMS/OS	AVERAGE	
SA	0	0	0	2000	5500	3750	$0+.004x10^6$/SA
UA	10000	8000	9000	5000	8500	6750	$.009x10^6+.007x10^6$/UA
FA	0	0	0	2500	8000	5250	$0+.005x10^6$/FA
FT	0	0	0	2500	4500	3500	$0+.004x10^6$/FT
WA/WS	5000	4000	4500	5000	7000	6000	$.005x10^6+.006x10^6$/WA-WS
WW	10000	24000	17000	10000	12000	11000	$.017x10^6x.011x10^6$/WW
NO	2000	4000	3000	2500	6000	4250	$.003x10^6+.004x10^6$/NO
TR	0	0	0	2500	4500	3500	$0+.004x10^6$/TR
TS	0	0	0	2500	4000	3000	$0+.003x10^6$/TS
GRID WIND							

TABLE B-5

SAMPLE ROUTES FOR HUB SIZING

ROUTE	SA	UA	FT	NOTAMS	WS	WA	DISTANCE
JAX/MIA	13	0, 1	7, 7	8, 4	0, 1	0, 1	300+
MSP/DET	11	1, 0	9, 9	5, 10	1, 0	2, 2	450+
SAN/LGB	7	0, 0	3, 3	0, 0	0, 0	0, 3	100+
DEN/J52/ATL	25	0, 6	10, 9	20, 16	0, 1	2, 9	1250+
EWR/TTN	3	0, 0	1, 1	1, 0	1, 1	0, 1	---
DCA/JFK	17	0, 1	8, 6	13, 5	1, 3	1, 3	450+
JAX/ATL	8	1, 3	3, 3	4, 5	0, 1	0, 1	250+
BAL/IAD/SBY	9	0, 2	4, 4	4, 5	0, 2	1, 2	150
AVERAGE	11.6	0.9	5.4	6.2	0.8	1.8	369+

TABLE B-6

WINDS AND TEMPERATURES ALOFT

TIMING DATA

PDP 11/70, RSX 11D

SINGLE ROUTE BOS/BAL

	CONSTANT TIME (SEC)	POINT TIME (SEC)	NUMBER OF POINTS
	0.0140	0.1484	8
	0.0153	0.1597	8
	0.0162	0.1653	8
	0.0160	0.1654	8
	0.0160	0.0207	1
	0.0155	0.0200	1
	0.0173	0.0195	1
	0.0154	0.0198	1
TOTAL	0.1257	0.7188	36
AVERAGE	0.0157	0.0200	

ASSUME MACHINE = $.25 \times 10^6$ INST./SEC

INSTRUCTIONS = $(0.25 \times 10^6)(0.0157 + 0.02/\text{POINT})$

= $(.004 \times 10^6 + .005/\text{POINT})$

three accommodates one SA and allows space for two potential SPs. Flags in the SA record for a LOCID indicate the presence of SPs, thus no attempt will be made to read SPs if none are available.

B. Access Technique - Binary search of SALOC yields ordinal of SA logical record.

Logic Disk Address = 3 * Ordinal - 2

C. Overhead accesses - None

D. Access for logical read - 1 of 256 bytes for SA

E. Overhead instruction count - 0

F. Access instruction count - 2000

B.2.1.2 UA

A. File Organization - Index record at file head, logical records follow.

B. Access technique - Read index into main memory and examine for LOCID match, if match, retrieve UA using logical disk address from index.

C. Overhead accesses - 1 of 3840 bytes to read index.

D. Access for logical record - 1 of 256 bytes for UA.

E. Overhead instruction count - 10000

F. Access instruction count - 5000

B.2.1.3 FA

A. File Organization - Records organized according to geographic region.

B. Access technique - Binary search of SALOC. If match on LOCID, byte 3 of SALOC entry contains geographic region code for LOCID. Logical Disk Address = geographic region code.

C. Overhead Accesses - None

D. Access for logical record - 1 of 4000 bytes for FA.

E. Overhead instruction count - 0

F. Access instruction count - 2500

B.2.1.4 FT

A. File Organization - Logical records arranged alphabetically to correspond to PSBCMI array FTLOC.

B. Access Technique - Binary search of FTLOC yields FTLOC ordinal which is the requried logical disk address.

C. Overhead Accesses - None.

D. Access for Logical record - 1 of 1024 bytes for FT.

AD-A047 930 MITRE CORP MCLEAN VA METREK DIV F/G 9/2
DISTRIBUTED PROCESSING APPLIED TO THE FLIGHT SERVICE STATION MO--ETC(U)
AUG 77 T B FOWLER DOT-FA69NS-162
UNCLASSIFIED MTR-7576 FAA/RD-77-161 NL

3 OF 3
AD
A047930

END
DATE
FILMED
1 -78
DDC

E. Overhead instruction count - 0

F. Access instruction count - 2500

B.2.1.5 WA/WS

A. File Organization - Doubly subscripted index table at head of file allows for ten WA/WS to be retrieved for each of the nine geographic regions.

B. Access technique - Read index into main memory, perform binary search of SALOC, extract geographic region, retrieve all available reports for geographic region - logical disk addresses available from index.

C. Overhead accesses 1 of 540 bytes for WA
1 of 540 bytes for WS

D. Access for logical record - 1 of 800 bytes for WA
1 of 800 bytes for WS

E. Overhead instruction count WA - 5000
WS - 5000

F. Access instruction count WA - 5000
Ws - 5000

B.2.1.6 WW

A. Index at head of file

B. Access technique - Read index into main memory, retrieve all reports, logical disk addresses available in index.

C. Overhead Accesses - 1 of 300 bytes

D. Access for logical record - 1 of 4096 bytes

E. Overhead instruction count - 10000

F. Access instruction count - 10000

B.2.1.7 NO

A. File Organization - Thirteen indexes for report collectives distributed every 130 logical records throughout file.

B. Access technique - Read thirtee index records into main memory, consolidate index, removing null entries, search sequentially for match on LOCID.

C. Overhead Accesses - 13 of 320 bytes

D. Access for logical record - 1 of 256 bytes

E. Overhead instruction count - 45000

F. Access instruction count - 8000

B.2.1.8 TR

A. File Organization - Indexed directly from TWEB Route Number

B. Access Technique - Convert TWEB route number to integer which is logical disk address.

C. Overhead Accesses - None

D. Access for logical record - 1 of 800 bytes for TR

E. Overhead instruction count - 0

F. Access instruction count - 2500

B.2.1.9 TS

A. File Organization - Data indexed by table internal to MEPTS, PFA

B. Access technique - Scan local table for LOCID match, table ordinal on LOCID match provide logical disk address

C. Overhead accesses - None

D. Access for logical record - 1 of 800 bytes for TS

E. Overhead instruction count - 0

F. Access instruction count - 2000

B.2.2 Route Processing Instructions Executed

Data for Table B-7 were gathered in a series of runs on MITRE's experimental route processor programs in a sterile environment on a PDP 11/70. No other users were permitted on the system. For

each route shown, the data in Columns A, B, C, D, and F were supplied by the computer through timing routines in the route processor inserted for that purpose. The other Columns were computed as shown from the computer-supplied information. The purpose of this study was threefold: to determine (1) system overhead (shown as percent of total route processor time, Column H), (2) number of instructions required for route processing (shown in Column J), and (3) any significant relationships between distance and number of instructions. The relationship between data in Columns J and K turned out to be the most important; a regression analysis of the numbers in these columns led to the following equation:

$$i = 348K + 180d$$

where i = number of instructions required for route processing (Column J) and d = route distance in nautical miles (Column K). This line and the data from Columns J and K are plotted in Figure B-1. The fit is very good, as demonstrated by the high value of the correlation coefficient from the regression analysis: 0.98.

The graph may be used for Hub sizing merely by determining the average length of a route of flight, and reading the corresponding number of instructions required directly.

Based on discussions with FSS Specialists at the Leesburg Center, the following data for the length of route oriented briefings appear

to be conservative estimates:

DISTANCE RANGE (nm)	PERCENT OF BRIEFINGS
100-500	85
>1500	15

Assuming a uniform distribution of lengths within each range, and a maximum length of 2500 nm, we can compute the average route length as follows:

$$E(d) = \left(\frac{500-100}{2} + 100\right) \times .85 + \left(\frac{2500-500}{2} + 500\right) \times .15 = 480$$

This figure does not reflect any briefings for routes less than 100 miles, since the Specialist said these were virtually never route-oriented. This figure, substituted into the above equation, leads to a mean number of instructions of

$$348K + (180)(420) = 424K$$

TABLE B-7

PDP-11/70 ROUTE PROCESSING DATA VS. ROUTE

ROUTE	A AVG TIME/ RLD CALL (MSEC)	B TOTAL CALLS TO RLD FROM RLT	A x B RLD + OVERHEAD (SEC)	C TOTAL TIME FOR ROUTE PROC. (SEC)	D TOTAL TIME IN OMSIO (DISK ACCESS TIME) (SEC)	E OVERHEAD TIME AXB - (D + B) X 2 X 10^{-4}* (SEC)
IAD/JFK	4.46	594	2.65	3.43	.866	1.66
BAL/DFW/SAN	3.52	2481	8.73	10.3	.851	7.39
LAX/V16/JFK	3.39	2953	10.01	11.5	1.03	8.39
BAL/SBY/EWR BOS/DET/MKC/PHX	3.35	2225	7.45	9.08	.829	6.18
DEN/J52/ATL	3.73	1954	7.29	8.20	1.05	5.85
TOS/J11/SLC	4.91	578	2.84	3.42	1.02	1.70
PVD/BOS	8.87	163	1.45	1.83	.851	.56

ROUTE	F DISK ACCESSES	G AVG TIME/ DISK ACCESS D/F (MSEC)	H % OF ROUTE PROC. TIME FOR OVERHEAD (E/C) X 100	I ROUTE PROCESSOR LESS OVERHEAD C - .01 X H (SEC)	J AVG. NO OF INSTR. @ 250,000/ SEC I X 2.5 X 10^5	K DISTANCE (n.m.)	L AVG. NO. INSTR. PER 100 NM
IAD/JFK	10	86.6	49	1.75	437.5K	200	2188
BAL/DFW/SAN	10	85.1	72	2.88	720K	2000	360
LAX/V16/JFK	11	93.6	73	3.11	777.5K	2500	311
BAL/SBY/EWR BOS/DET/MKC/PHX	10	82.9	68	2.91	727.5K	2250	323
DEN/J52/ATL	11	95.5	71	2.38	595K	1250	476
TOS/J11/SLC	11	92.7	50	1.71	427.5K	600	712
PVD/BOS	10	85.1	31	1.26	315K	80	3938

*ASSUMES 50 MACHINE INSTRUCTIONS (4 SEC EACH AVERAGE) EXECUTED FOR EVERY ENTRY INTO RLDX PROGRAM (EXLUDING SUBROUTINE AND SYSTEM CALLS).

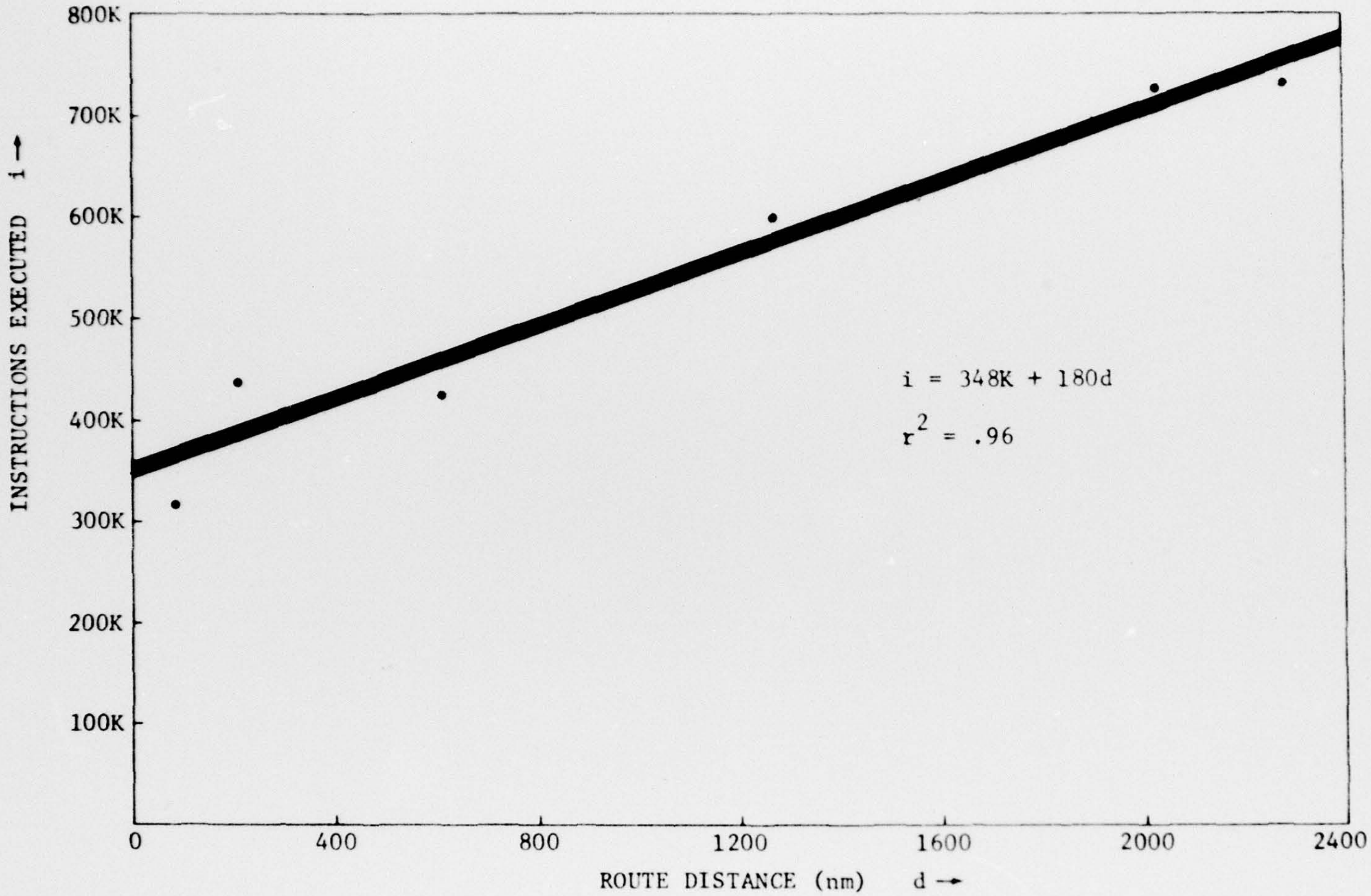


FIGURE B-1
INSTRUCTIONS RQD FOR ROUTE PROCESSING VS ROUTE DISTANCE

APPENDIX C

RESPONSE TIME ESTIMATES FOR LARGE MAINFRAME COMPUTER SYSTEMS

To facilitate evaluation of the response times for minicomputer systems derived in Section 4, similar calculations are made in this Appendix for two sample large-scale computer systems, assuming 1995 peak load. Such a system would of necessity be of the load sharing type, with either one machine handling the entire load, and one as a 'hot standby,' or two machines sharing the load equally, but either capable of taking over completely in the event of a failure in the other. The procedure followed in Section 4.2 for determining expected waiting time and expected response times is also employed here, with the exception that only one case is analyzed, viz. the 1995 peak load. The same disk access times are also assumed.

Table C-1 summarizes expected waiting times and related queueing parameters for a system with raw computing speed of 2.4 MIPS, which (applying the same factor as in Section 4.2) is assumed to be equivalent to 1.5 MIPS of available speed. Table C-2 presents the corresponding mean response times for the seven categories of FSS activity, and may be compared directly with Tables 4-25 and 4-26. Such a system would be a UNIVAC 1100/40 multiprocessor, costing between $3M to $5M, or a Burroughs B6800 multiprocessor, costing between $3M to $4M.*

---

* Estimates based on MITRE Working Paper WP-12218, <u>FSS Hub Sizing and Costing</u>, by M. I. Gasper. The current costs may be lower due to the recent round of price cuts.

Similarly, Table C-3 summarizes expected waiting times for a machine with an available speed of about 1.1 MIPS, and Table C-4 gives the corresponding mean response times, which again may be compared directly with Tables 4-17 and 4-28. Such a system would be a single 1100/40.

TABLE C-1

MEAN WAITING TIMES FOR 1.5 EQUIVALENT MIPS CONFIGURATION

TYPE	INSTR	α_1	α_2	λ PER SECOND	E(W)
1. Processor Status Messages (Constant Service Time)	250	1.67×10^{-4}	2.78×10^{-8}	5	.0002
2. Data Base Update (E_3 Service Time)	47K	.0313	.0013	1	.0320
3. Input from Terminals* (E_3 Service Time)	10K	.0067	5.93×10^{-5}	1.25	.0077
4. Output to Terminals, Wx Data (E_{10} Service Time)	5K	.0033	1.01×10^{-5}	2.69	.0042
5. Output to Terminals, Other Categories (E_2 Service Time)	1K	.0007	6.6×10^{-7}	.833	.0015
6. I/O Ready List, Disk Processing Weather Data for First Page (E_3 Service Time)	16K	.0107	.0002	.253	.0121
7. Flight Plan Processing (Service Time Parameters as determined in Section 4.1.4.4.2)	86K	.0573	.0621	.115	.0653
8. Aircraft Contact Processing (E_3 Service Time)	20K	.0133	.0002	.0525	.0190
9. Miscellaneous Messages** (E_2 Service Time)	9500	.0063	6.02×10^{-5}	6.67	.0120
10. Route Processing (E_{10} Service Time)	434K	.2893	.0921	.1683	.3382
11. I/O Ready List, Rest of Route Brief (E_2 Service Time)	671K	.4473	.300	.1683	.5837
12. I/O Ready List, Rest of Local Brief (E_3 Service Time)	76K	.0507	.0034	.0850	.1287

* Assumes 4500 per hour, based on 1500 weather and flight plan nessages, and 2 x 1500 = 3000 miscellaneous messages.

** Includes 4500 logging messages and recording of incoming data for a total of 16,517 + 3000 + 4500 = 24,017 messages per hour.

TABLE C-2

MEAN RESPONSE TIMES FOR CATEGORIES OF FSS ACTIVITY
(1.5 EQUIVALENT MIPS CONFIGURATION)

1.	ROUTE BRIEF (ITEMS 3, 4, 6, 10)	.362 seconds
	DISK ACCESSES	.656
		1.02
2.	LOCAL BRIEF (ITEMS 3, 4, 6)	.0240
	DISK ACCESSES	.131
		.155
3.	FLIGHT PLANS (ITEMS 5, 7)	.0668
	DISK ACCESSES	.263
		.330
4.	AIRCRAFT CONTACTS (ITEMS 5, 8)	.0205
	DISK ACCESSES	.105
		.126
5.	MISCELLANEOUS (ITEMS 5, 9)	.0135
	DISK ACCESSES	.131
		.145
6.	REST OF ROUTE BRIEF (ITEM 11)	.584
	DISK ACCESSES	1.838
		2.42
7.	REST OF LOCAL BRIEF (ITEM 12)	.129
	DISK ACCESSES	.656
		.785

TABLE C-3

MEAN WAITING TIMES FOR 1.1 EQUIVALENT MIPS CONFIGURATION

TYPE	INSTR	α_1	α_2	λ PER SECOND	E(W)
1. Processor Status Messages (Constant Service Time)	250	2.35×10^{-4}	5.54×10^{-8}	5	.0002
2. Data Base Update (E_3 Service Time)	47K	.0442	.0026	1	.0456
3. Input from Terminals (E_3 Service Time)	10K	.0094	.0001	1.25	.0114
4. Output to Terminals, Wx Data (E_{10} Service Time)	5K	.0047	2.44×10^{-5}	2.69	.0066
5. Output to Terminals, Other Categories (E_2 Service Time)	1K	.0009	1.33×10^{-6}	.833	.0026
6. I/O Ready List, Disk Processing, Weather Data for First Page (E_3 Service Time)	16K	.0151	.0003	.253	.0179
7. Flight Plan Processing (Estimated Service Time Parameter)	86K	.0809	.0847	.115	.0948
8. Aircraft Contact Processing (E_3 Service Time)	20K	.0188	.0005	.0525	.0280
9. Miscellaneous Messages (E_2 Service Time)	9500	.0089	.0001	6.67	.0182
10. Route Processing (E_{10} Service Time)	434K	.4085	.1835	.1683	.5100
11. I/O Ready List, Rest of Route Briefing (E_2 Service Time)	671K	.6100	.5582	.1683	.9030
12. I/O Ready List, Rest of Local Briefing (E_3 Service Time)	76K	.0691	.0064	.0850	.2502

TABLE C-4

MEAN RESPONSE TIME FOR CATEGORIES OF FSS ACTIVITY
(1.1 EQUIVALENT MIPS CONFIGURATION)

1.	ROUTE BRIEF (ITEMS 3, 4, 6, 10)	.5459 seconds
	DISK ACCESSES	.656
		1.20
2.	LOCAL BRIEF (ITEMS 3, 4, 6)	.0359
	DISK ACCESSES	.131
		.167
3.	FLIGHT PLANS (ITEMS 5, 7)	.0974
		.263
		.360
4.	AIRCRAFT CONTACTS (ITEMS 5, 8)	.0306
	DISK ACCESSES	.105
		.136
5.	MISCELLANEOUS (ITEMS 5, 9)	.0208
	DISK ACCESSES	.131
		.152
6.	REST OF ROUTE BRIEF (ITEM 11)	.9030
	DISK ACCESSES	1.838
		2.74
7.	REST OF LOCAL BRIEF (ITEM 12)	.2502
	DISK ACCESSES	.656
		.906